JC18 Rec'd PCT/PTO 17 DEC 2001

FORM PTO-1390 REV. 2/01T — U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER: 08364.0031

U.S. APPLICATION NO. (If known, see 37CFR1.5): 10/009722

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/GB00/02336	June 15, 2000	June 18, 1999

TITLE OF INVENTION
METHOD AND APPARATUS FOR MONITORING AND MAINTAINING THE CONSISTENCY OF DISTRIBUTED DOCUMENTS

APPLICANT(S) FOR DO/EO/US
Andrea ZISMAN; Anthony FINKELSTEIN, Ernst ELLMER, Danila SMOLKO, and Wolfgang EMMERICH

Applicant(s) herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☐ This is an express request to begin national examination procedures (35 U.S.C. 371(f)). The submission must include items (5), (6), (9) and (21) indicated below.
4. ☒ The US has been elected by the expiration of 19 months from the priority date (Article 31).
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c)(2)).
   a. ☒ is attached hereto (required only if not communicated by the International Bureau.
   b. ☒ has been communicated by the International Bureau.
   c. ☐ is not required, as the application was filed with the United States Receiving Office (RO/US).
6. ☐ An English language translation of the International Application as filed (35 U.S.C. 371 (c)(2)).
   a. ☐ is attached hereto.
   b. ☐ has been previously submitted under 35 U.S.C. 154 (d)(4).
7. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3)).
   a. ☐ are attached hereto (required only if not communicated by the International Bureau).
   b. ☐ have been communicated by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☒ have not been made and will not be made.
8. ☐ An English language translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371 (c)(3)).
9. ☐ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
10. ☐ An English language translation of the annexes of the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).

Items 11 to 20 below concern document(s) or information included:

11. ☒ Information Disclosure Statement under 37 CFR 1.97 and 1.98
12. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. ☐ A FIRST preliminary amendment.
14. ☐ A SECOND or SUBSEQUENT preliminary amendment.
15. ☐ A Substitute specification.
16. ☐ A change of power of attorney and/or address letter.
17. ☐ A computer-readable form of the sequence listing in accordance with PCT Rule 13ter.2 and 35 U.S.C. 1.821-1.825.
18. ☐ A second copy of the published international application under 35 U.S.C. 154 (d)(4).
19. ☐ A second copy of the English language translation of the international application 35 U.S.C. 154 (d)(4).
20. ☒ Other items or information:
   a. ☒ Copy of cover page of International Publication No. WO00/79428 A2
   b. ☐ Copy of Notification of Missing Requirements.
   c. ☐

10/009722 JC18 Rec'd PCT/PTO 17 DEC 2001

531 Rec'd PCT/ 17 DEC 2001

U.S. APPLICATION NO. (If known, see 37 CFR 1.5) 10/009722	INTERNATIONAL APPLICATION NO. PCT/GB00/02336	ATTORNEY'S DOCKET NUMBER 08364.0031

21. ☒ The following fees are submitted:

BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)):

Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO **$1040.00**

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO **$890.00**

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search fee (37 CFR 1.445(a)(2)) paid to USPTO **$740.00**

International preliminary examination fee (37 CFR 1.482) paid to USPTO but all claims did not satisfy provisions of PCT Article 33(1)-(4) **$710.00**

International preliminary examination fee (37 CFR 1.482) paid to USPTOand all claims satisfied provisions of PCT Article 33 (1)-(4) **$100.00**

	CALCULATIONS PTO USE ONLY	
ENTER APPROPRIATE BASIC FEE AMOUNT =	$890.00	
Surcharge of $130.00 for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (e)).	$	

CLAIMS	NUMBER FILED		NUMBER EXTRA	RATE		
Total Claims	48	- 20 =	28	x $18.00	$504.00	
Independent Claims	5	-3 =	2	x $84.00	$168.00	
MULTIPLE DEPENDENT CLAIM(S) (if applicable)				+$280.00	$280.00	
				TOTAL OF THE ABOVE CALCULATIONS =	$1842.00	
☐ Applicant claims small entity status. See 37 CFR 1.27. The fees indicated above are reduced by ½.					$	
				SUBTOTAL =	$1842.00	
Processing fee of **$130.00** for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest priority date (37 CFR 1.492(f)).					$	
				TOTAL NATIONAL FEE =	1842.00	
Fee for recording the enclosed assignment (37 CFR 1.21 (h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). **$40.00** per property. +					$	
				TOTAL FEES ENCLOSED =	$1842.00	
				Amount to be refunded:	$	
				charged:	$	

a. ☒ A check in the amount of $ 1842.00 to cover the above fees is enclosed.

b. ☐ Please charge my Deposit Account No. _____ in the amount of $ _____ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 06-0916. A duplicate copy of this sheet is enclosed.

d. ☐ Fees are to be charged to a credit card. **WARNING:** Information on this form may become public. **Credit card information should not be included on this form.** Provide credit card information and authorization on PTO-2038.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137 (a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.
1300 I Street, N.W.
Washington, D.C. 20005-3315

DATED: December 17, 2001

SIGNATURE

Ernest F. Chapman, Reg. No. 25,961

NAME/REGISTRATION NO.

10009722 122701

10/009722
16/PRTS 531 Rec'd PCT/ 17 DEC 2001

METHOD AND APPARATUS FOR MONITORING AND MAINTAINING THE CONSISTENCY OF DISTRIBUTED DOCUMENTS

Technical Field

This invention relates to methods and apparatus for monitoring and maintaining the consistency of documents having related overlapping content, for example documents in which the content is represented using XML (eXtensible Markup Language) and other related technologies.

Background to the Invention

Documents can either be created by individuals or by a group of individuals. Different documents created can be related or have overlapping content in that they relate to common objects and phenomena. In such circumstances, required relationships between data in one document and data in the same or other documents may be formalised as a set of consistency rules defining those relationships.

An example of a simple consistency rule might be that data in one document which also appears in another document is considered to relate to the same object or phenomena. More complex formal relationships can exist within documents such as design specification documents which are intended to contain data which is represented

in different documents in different ways in a pre-defined manner.

In circumstance where new documents, document types and consistency rules may be added at any time it becomes difficult to ensure that, during the production of a set of documents, the set of documents fulfils all of the requirements of a formal set of consistency rules at relevant key points within the production process. This is particularly the case since during the production of a set of documents, it is inevitable that on occasions some consistency rules will not be satisfied. Thus for example where it is required that data present in one document also appears in another, when data is initially entered for the first document the other document may not immediately reflect the change made to the first document.

Where documents are generated in distributed organisations by large numbers of individuals it becomes impractical to limit the production of documents to ways which ensure as far as is possible that the consistency of a set of documents is constantly maintained. Also, with large sets of documents it becomes very difficult for formal consistency relationships to be checked

SCANNED, # 10

manually as searching through the data of a large set of documents is time consuming and errors occur.

The present invention aims to provide a means by which data which is intended to comply with formal consistency relationships defined by a set of consistency rules can be monitored whilst tolerating the entry of data which does not fulfil the required relationships.

In accordance with one aspect of the present invention there is provided a computer apparatus comprising:

means for generating a first set of structured documents containing data;

means for defining a set of consistency rules defining relationships which are intended to be fulfilled by the data within the first set of documents; and

means for generating a second set of documents containing data of said first set of documents, wherein the data of the second set of documents is linked on the basis of whether said data does or does not fulfil the relationships defined by said consistency rules.

By providing means by which a linked set of documents can be generated, users are then able to view data which either fulfils or does not fulfil a consistency rule in

the context of the document where that data appears.

Disclosure of the Invention

In accordance with another aspect of the present invention there is provided computer apparatus comprising:

means for inputting documents containing data;

means for inputting a set of consistency rules defining relationships which should be fulfilled by data in said documents; and

means for generating consistency links identifying data in said documents which does and does not fulfil the formal relationships defined by said consistency rules.

Embodiments of the present invention enable the documents and generated consistency links to be viewed by a user in a number of different ways in order to help a user identify why data within a set of documents does or does not fulfil a defined set of relationships.

Brief Description of the Drawings

Embodiments of the present invention will now be described by way of example with reference to the accompanying drawings in which:

Figure 1 is a block diagram of a computer network embodying the present invention;

Figure 2 is an example of a set of design documents having overlapping content for which a consistency report could be generated;

Figure 3 is a diagrammatic representation illustrating an example of a consistency report generated for the set of design documents of Figure 2 comprising a set of linked documents with data within the documents linked in accordance with a formal relationship defined by an exemplary consistency rule.

Figure 4 is another example of a consistency report generated for the design documents of Figure 2 comprising a report identifying data which does and does not comply with a formal relationship defined by an exemplary consistency rule;

Figure 5 is a block diagram of the memory of the apparatus of Figure 1 having a consistency checker showing the programs and data initially stored in the memory of the apparatus;

Figure 6 is a block diagram illustrating the data structure for a consistency rule defined by a consistency rules DTD.

Figure 7 is a block diagram of further data down loaded or generated by the apparatus of Figure 1 having a consistency checker and stored in the memory of the apparatus;

Figure 8 is a flow diagram of the steps involved in the generation of an output presentation;

Figure 9 is a flow diagram illustrating the generation of a set of consistency rules using a consistency rule editor;

Figure 10 is a flow diagram illustrating the steps involved in the addition of a consistency rule to a set of consistency rules using a consistency rule editor;

Figure 11 is a flow diagram illustrating a generation of consistency links using a consistency link generator;

Figure 12 is a flow diagram illustrating the generation of CT consistency links;

10/089722 052202

Figure 13 is a block diagram of a data structure of a consistency link.

Figure 14 is a flow diagram illustrating the generation of CF consistency links;

Figure 15 is a flow diagram illustrating the generation of IF consistency links;

Figure 16 is a flow diagram illustrating the processing of a watch dog program; and

Figure 17 is a flow diagram illustrating the processing of a consistency manager program.

FUJ09722-052202

Embodiments

In the following description, the present invention is described in terms of an XML (eXtensible Markup Language) implementable software system. XML is a data description language which is a sub set of SGML (Standard Generalised Markup Language) which has been standardised by the Worldwide Web Consortium (W3C). XML has been generated in view of the main problems involved in hard coding presentation information and linking information into documents using HTML (Hyper Text Markup Language).

In contrast to HTML in which content, presentational information and link information are all encoded in a single document, in XML the structure of a document, how it is linked to other documents and how it is presented to a user can be separated.

In XML, before a document is created a DTD (Document Type Definition) is created which defines a data structure for a document. This comprises a set of rules defining how documents and tags are to be structured, which elements are presented and the structural relationship between the elements for documents of a particular type.

A very simple example of a document type definition for a book might be that a book comprises one or more pages, each page having one or more paragraphs, each paragraph comprising one or more sentences, each of the sentences comprising a number of words and ending with a full stop. Documents of the data type book would then be instances of data structured in the manner determined by the book DTD.

XML also provides a means by which different parts of

data structured in accordance with DTD can be linked to one another, known as XML linking or X link. An X link comprises a definition of where data which is to be linked is located. An X link defines the location of data in terms of a computer file for a structured document and where within that structured document the data is to be found. The structure of an X Link can itself be defined in XML by a DTD as consisting of two parts, a URL (Universal Resource Locator) identifying the file containing data which is to be linked and an optional X-pointer expressing where within a document structured in accordance with a DTD the link is to point to. Thus, for a document defined in terms of the above simple book DTD example, an X Link would comprise a URL identifying a particular book document and an X-pointer to a particular word in that book in terms of which page, paragraph, sentence and where within a sentence that word appears.

The presentation of an output XML document is defined using XSL (eXtensible Style Language). XSL is used to define description of the mapping between the content elements of an XML document and presentation elements which are to be output. An XSL file describing a mapping from content elements to presentation elements is known

as an XSL style sheet. An output is then generated by using an XSL style sheet processor which processes XML documents to generate an output by applying the mapping defined by an XSL style sheet. In this way by providing different XSL style sheets an XML document can be translated between different formats. Thus an XSL style sheet may define how an XML document is to be displayed in for example HTML (Hyper Text Markup Language) so that an XML document may be viewed within a standard browser.

An embodiment of the present invention will now be described in which Figure 1 is a block diagram of a distributed computer network embodying the present invention. In this embodiment the distributed computer network comprises a plurality of computer terminals 1, 2, 3, 4 connected to each other via a communications network 5. Each of the computer terminals 1-4 comprises a computer 6 that is connected to a display 7. Also connected to each computer 5 is a keyboard 8 and a mouse 9.

Stored within the memory of one of the computer terminals 4 of the network of Figure 1 is a consistency checker program in accordance with the present invention. The consistency checker program 10 comprises a link data

generator 11, a report generator 12 and a monitor program 13 as will be described in detail later. Also stored in the memory of the computer 6 is a set of consistency rules 14 defining a set of relationships that data in documents are required to fulfil as will also be described below. Further programs and data (not shown in Figure 1) are also stored in the memory of the computer 6 as are detailed later.

In this embodiment of the present invention, stored within the memory of the computers 6 of all of the terminals 1-4 is application software 15 for generating XML (eXtensible Markup Language) source documents for UML (unified modelling language) software design diagrams, using the keyboard 8 and the mouse 9 of the terminals 1-4. The application software 15 is arranged to enable users to generate UML design documents and to store these documents in the form of XML source documents in the memory of the computer 6 of terminals 1-4 on which they are located.

In accordance with the present embodiment, users generate a set of XML source documents for UML software design diagrams using the application software 15 on each of the terminals 1-4. These XML source documents are then

transferred via the communications network 5 to the terminal 4 of the distributed computer network having stored in its memory the consistency checker program 10. The XML source documents are then processed by the link data generator program 11 in accordance with the set 14 of consistency rules to generate a set of XML consistency link elements, representing links between data in the source documents linked in accordance with the consistency rules 14. The XML source documents, XML consistency link elements and consistency rules are then processed by the report generator 12 to generate HTML output documents including HTML links so that data within the set of output documents may be navigated by a user so as to view data within the documents linked in accordance with whether the data in the documents does or does not fulfil the formal relationships defined by the consistency rules.

The processing of the consistency checker 10 will now be described with reference to Figure 2 which illustrates an exemplary set of UML design documents that might be generated using the application software 15 stored in the computers 6 of the terminals 1-4 of the network of Figure 1. This exemplary set of design documents comprises a business entities class diagram 1000 and a create meeting

collaboration diagram 2000 for the design of software for a simple meeting scheduler system.

In this set of documents, the business entities class diagram 1000 illustrates the interaction of classes in the meeting software which is to be created in accordance with the design. Each of the classes represents data structures that can be fulfilled by specific instances of data created during the course of the use of the meeting scheduler software. The classes in this example comprise a user class 1001, a message class 1002, a meeting class 1003 a location class 1004, a date class 1005, an organiser class 1006, and a participant class 1007. The business entities class diagram 1000 also illustrates how the different classes relate to each other. Thus in this example the user class 1001 is a data structure for each user comprising a password, name and e-mail address. The participant 1007 and user 1006 classes are sub-classes of the user class 1001 and have the same data structure. The message class 1002 is a data structure for message data which relates to a message sent between users about meetings. The meeting class 1003 is a data structure for data about the name and status of a meeting and which is related to messages about a meeting, an organiser, one or more participants, a date and a location. The date and

location merely in terms of a date and a time period and a location for a meeting respectively.

The create meeting collaboration diagram 2000 of the illustrated set of documents is a diagram illustrating the steps and interactions which occur when a specific meeting is to be set up. The create meeting collaboration diagram 2000 in this example includes a set 2001-2004 of instances representing the specific examples of data of certain types which interact when a meeting is set up using the proposed software in accordance with a design. The instances in this example comprise an organiser window 2001, an organiser 2002, a meeting 2003 and a date 2004. Thus in this example an organiser window is used which generates meeting data which is sent to an organiser and causes a calender to be updated.

After the set of design documents 1000,2000 has been generated using the application software of 15 and the XML corresponding to the those documents has been generated they are made available to the consistency checker 10 via the communications network 5. The consistency checker 10 is then able to generate consistency links in accordance with a set of consistency rules 14 stored in memory defining required relationships

between data in the documents created.

An example of the processing by the consistency checker will now be described in which we consider the processing of a set of consistency rules 14 comprising a single consistency rule 14 defining a relationship that data within UML design documents are required to fulfil. The exemplary consistency rule which is considered is a rule that: "For every instance in a collaboration diagram there must exist a class in a class diagram of the same name". This rule defines a formal requirement of a UML design document that where a specific instance of data is shown as a specific type interaction in a collaboration diagram the general overview of how data of that type interacts with data of other classes should also be represented in the set of design documents.

When processing a set of consistency rules, the consistency checker 10 initially invokes the link data generator 11 which identifies from the XML source for the design documents 1000,2000 all data in the collaboration diagram 2000 which correspond to instances 2001-2004 in the collaboration diagram 2000 as data which is required to fulfil the conditions of the consistency rule 14. The link data generator 11 then identifies all of the data in

the business entities class diagram 1000 which corresponds to classes, as potential data which may satisfy the condition of the consistency rule. The way in which the data corresponding to classes and instances is identified will be described in detail later. Both of the sets of identified data are then stored in memory in the computer 6 of the terminal 4.

The link data generator 11 then generates a set of XML link elements (X links) corresponding to consistency relations in the UML design documents 1000,2000 using the consistency rules. This is achieved by the link data generator 11 testing each piece of data identified as possibly relevant against the condition of the consistency rule under consideration, and then generating appropriate link elements depending upon the outcome of the tests.

Thus in this example the link data generator 11 would test all of the instances 2001-2004 in the collaboration diagram 2000 to see if any class data 1001-1007 in the business entity diagram 1000 has the same name. Whenever data within the collaboration diagram 2000 is matched with data from the class diagram 1000, link elements are generated identifying the two pieces of data and a status

is assigned to the link to indicate that these pieces of data fulfil the requirements of the consistency rule under consideration. If data from the collaboration diagram 2000 cannot be matched, a link element is generated labelling the data as inconsistent with respect to the consistency rule under consideration.

Thus in the present example the following XML link elements would be generated.

Data from Collaboration Diagram 2000	Data from Class Diagram 1000	Status
Organiser Window Instance 2001	None	Inconsistent
Organiser Instance 2002	Organiser class 1006	Consistent
Meeting Instance 2003	Meeting class 1003	Consistent
Date Instance 2004	Date class 1005	Consistent

The XML link elements, consistency rules 15 and source documents are then passed to the report generator 12 to generate an output. In this embodiment the presentation report generator 12 is composed of a style sheet processor and a set of construction rules. The construction rules are applied to the XML source documents, consistency rules and XML link elements using the style sheet processor to generate HTML (Hypertext

Markup Language) documents.

In the current example the HTML documents generated by the report generator 12 would be HTML representations of the XML source documents for the UML design diagrams 1000,2000. In addition these HTML documents also include HTML links between data identified by the XML link elements generated by the link data generator 11 which is or is not consistent with the set of consistency rules and HTML labels labelling data as consistent or inconsistent.

Figure 3 is a diagrammatic representation illustrating a consistency report for the set of documents of Figure 2. Figure 3 shows a set of linked HTML documents 1000,2000 comprising HTML representations of the documents 1000,2000 of Figure 2 linked by HTML links in accordance with the consistency rule that for every instance in a collaboration diagram there must exist a class in a class diagram of the same name.

The links in a generated consistency report comprise HTML links to other data. The HTML links appear within the output document and cause the data identified by a link

to be displayed when the link is selected. In Figure 3 the HTML links are shown as lines connecting data in different documents. In this example the HTML links comprise a link 3001 between the organiser instance 2002 of the create meeting collaboration diagram 2000 and the organiser class 1006 of the business entities class diagram 1000, a link 3002 between the meeting instance 2003 of the create meeting collaboration diagram 2000 and the meeting class 1003 of the business entities class diagram 1000, and a link 3003 between the date instance 2004 of the create meeting collaboration diagram 2000 and the date class 1005 of the business entities class diagram 1000.

All of the data in the collaboration diagram linked to data in the class diagram is also labelled by an HTML label which is displayed when a user selects the linked data which indicates why the links have been generated. Thus in the present example all of the linked data 2002-2004, is labelled with an HTML label indicating that the data is consistent with the consistency rule that for every instance in a collaboration diagram there must exist a class diagram of the same name. An HTML label is a label which is displayed whenever the data which is labelled is selected. In Figure 3 the HTML labels

indicating that data is consistent with the consistency rule are shown as speech bubbles 3004 detailing the information which would appear when the labelled data is selected.

No link is generated to the organiser window 2001 to data in the business entities class diagram 1000 for which no corresponding class exists. However the data is labelled 3005 in HTML to indicate that the data is inconsistent with the consistency rule.

By generating a linked and labelled set of documents, a user can view data which has been linked within the context of each document and is aided in establishing why data is or is not consistent with the requirements of a set of consistency rules.

Figure 4 is an illustration of an alternative representation for the output presentation for the documents of Figure 2 in which the link elements and consistency rules have been processed in accordance with a different XSL style sheet. In this example the output presentation comprises a consistency report 4000. The consistency report 4000 comprises for each of the rules of the set of consistency rules 14 in memory, a rule

identification 4001, a definition of the rule 4002 and a list 4003-4006 of data to which the rule is applicable for which links have been generated together with labels as to whether the data is or is not consistent with the rule.

Thus, the consistency report for the documents of Figure 2 therefore comprises a rule identification 4001, a definition 4002 of that rule for every instance in the collaboration diagram there must exist a class in a class diagram of the same name and a set 4003-4006 of data corresponding to each of the instances 2001-2004 in the create meeting collaboration diagram to which this rule is applicable and a label indicating whether the link is generated for that instance of data are consistent or inconsistent with the condition for the rule identified by the identification data.

In this example the data 4003-4005 comprising meeting, date and organiser which fulfilled the requirements of this rule are labelled therefore as consistent and the data 3006 organiser window is labelled as inconsistent.

By providing a consistency report detailing which rules are and are not fulfilled by data in a set of documents

an overview of how well the documents fulfil consistency relationships defined by a set of consistency rules.

As a further alternative the XSL style sheet of the report generator 12 may be arranged to generate both the linked set of documents of Figure 3 and the report of Figure 4. In this case the report generator could be arranged so that items of data 4003-4006 appearing in the list in the report 4000 are linked via HTML links to the corresponding data within the linked set of documents so that a user may view both an overview report and data within its proper context to establish why consistency rules are or are not being fulfilled.

When an output presentation has been generated it is then necessary to ensure that the correspondence between the output presentation and the source documents is maintained. This is achieved by the monitor program 13. The monitor program 13 monitors the XML source documents to determine whether updated copies of the XML source documents have been made available on any of the computer terminals 1-4 in the distributed computer network. If an update of documents is detected the monitor program 13 is arranged to initiate the generation of a new set of links by the consistency link generator 11 and an output

presentation by the report generator 12 to account for the updates made to the source documents. In this way, once an output presentation is created the correspondence between output presentation and updated source documents is maintained.

A detailed description of the content of the memory of a computer terminal having a consistency checker in accordance with the present invention stored in its memory will now be described in greater detail with reference to Figures 5 to 7.

Figure 5 is a block diagram of programs and data for a consistency checker 10 as initially stored in the memory of the computer apparatus 4 of Figure 1. The programs and data 11 - 40 initially stored in the memory will now be described.

In addition to the link data generator 11, the report generator 12, the monitor program 13 and the design document creation program 15 also stored in the memory is an XML editor 20 which comprises a standard XML editor which enables a user to create and edit XML, DTD and XSL files. The XML editor also provides an interface for enabling a user to cause XML documents to be displayed on

the display 7 as text or as a tree structure, a DTD file to be displayed on the display 7 as text or as a tree structure and an XSL style sheet to be displayed on the display 7 as text or to display the output of processing of an XSL style sheet as text. In addition the XML editor also controls the loading of XML, DTD and XSL files into the memory so that they can then be edited using the keyboard 8 and mouse 9.

Also stored in the memory is a consistency rule DTD 30 which comprises a data file defining a data structure for consistency rules.

Figure 6 is a block diagram of the data structure defined by the consistency rules DTD 30 stored in the memory of the computer apparatus of Figure 4 having a consistency checker 10. The data structure defined by the consistency rules DTD 30 comprises: an identification tag 21, a text description of a consistency rule in a natural language 22, a rule type 23 as described below, X-pointers to source and destination elements 24, 25 identifying possible related elements in a set of documents and a set 26 of one or more conditions which represent the conditions of the rule to be satisfied in order to establish a consistency link between an instance

of the source element and an instance of the destination element.

Thus in the previous example of a consistency rule described in relation to figures 2, 3 and 4 the consistency rule, stored as data in accordance with his data structure would comprise: an identification tag 21 say for example "rule 1", a text description "For every instance in a collaboration diagram there must exist a class in an associated class diagram of the same name", a rule type 23, an X-pointer 24 to the part of the data structure of a collaboration diagram which is filled by data representing instances, an X-pointer to the part of the data structure of a class diagram which is filled by data representing classes and a condition that data found at the parts of the documents identified by the X-pointers 24,25 is required to fulfil namely that they are identical.

In this embodiment of the invention there are three rule types referred to as CT, CF and IF type rules corresponding to three different types of consistency relationships which are described below.

A CT rule is a type of rule which represents a

relationship between data in documents that ought to be fulfilled by all instances of data which are to be considered to be linked. A CT rule therefore is a rule which causes two instances to be labelled as being consistent whenever a consistency condition is fulfilled by a pair of data from those instances and for data to be labelled as inconsistent if a condition does not hold for data in a document compared with any of the other data in the set of source documents.

An example of a CT type rule is the consistency rule described above that there must be a class for every instance in a collaboration diagram. This would require that for all the instances of data in a collaboration diagram there must be an equivalent class. If the class does exist for an instance, the instance in the collaboration diagram would be required to be linked to the corresponding class and labelled as being consistent. If however, there is an instance in a collaboration diagram for which no class exists this instance would be required to be labelled as inconsistent as it does not fulfil the consistency rule.

A second type of rule is a CF rule which is a type of rule which represents a relationship which does not have

to be in existence but if it exists causes instances of data to be linked as consistent instances. A CF rule is therefore a rule which causes two elements to be linked if those elements fulfil a consistency relationship, but does not generate an inconsistent link if a consistency condition is not fulfilled.

An example of a CF rule for the generation of UML software design documents might be that classes with the same name in different class diagrams are considered to be identical. In this case where classes have the same name they would be required to be linked as being consistent. However there could be instances of classes that only exist in a single class diagram. The existence of such classes would not indicate that an inconsistency has occurred and hence no link would be required to be generated if such instances were to be detected.

The third type of rule type in this embodiment is an IF rule which is a type of rule which represents a relationship which must not be fulfilled by instances in associated documents. An IF rule is therefore a rule which causes two instances to be linked as being inconsistent whenever the condition of the rule is fulfilled by those instances, and for data to be labelled

consistent if no other data exists which fulfils the condition.

An example of an IF rule for a UML software design document might be a closure relationship such as that for every class C1 with a sub-class C2 in a class diagram, C2 must not be a super class of C1 in another diagram. If the condition of such a rule were to be fulfilled the two instances of the class C1 and C2 would be required to be linked as being inconsistent. If, however, the condition of the rule were not fulfilled the class C1 would need to be labelled as consistent but not be linked with any particular instance of class C2.

In addition to the XML editor 20 and the consistency rules DTD 30, as shown in Figure 5, also stored in the memory are a consistency rule editor 40 which is a program for generating and editing sets of consistency rules as will be described in detail later. The link data generator 11, report generator 12 and monitor program 13 each comprise more than a single module each of which are indicated separately in Figure 5. These modules will now be briefly described.

The link data generator 11 stored in memory comprises: a

document universe editor 50 which is a program for selecting which documents are to be processed against a given set of consistency rules; and a consistency link generator 60 which is a program for generating consistency link elements in accordance with consistency rules generated by the consistency rule editor 40.

The report generator 12 stored in memory comprises: an output presentation style sheet 70 which is an XSL style sheet for determining the presentation of an output report and a style sheet processor 80 which is an XSL processor for processing a set of XML documents, a set of consistency rules and a set of consistency link element in accordance the output presentation style sheet 70 to generate an HTML output presentation 170 which may be browsed using a standard browser.

The monitor program 13 stored in memory comprises a monitor module 90 which is a program for monitoring the modification of XML source documents for which a consistency report has been generated and a consistency manager 100 which is a program for updating an output report generated by the style sheet processor 80 to update an output report to account for changes made to a set of XML source documents.

Figure 7 is a diagram of data generated and stored in memory of the apparatus 4 of Figure 1 having a consistency checker in the course of generating an output presentation 170 in addition to the data and programs initially stored in the memory as shown in Figure 5.

Stored in the memory are a number of source document DTD's 110 defining the data structures for source documents. A number of source documents 120 structured in accordance with the source document DTDs 110 are also stored in the memory. The source documents 120 comprise XML documents for which an output presentation 170 is to be generated.

Thus in the above described example the source documents 120 would comprise XML source documents for the UML design documents 1000,2000 and the source documents DTD 110 would comprise document type definitions for the business entities class diagram 1000 and the collaboration diagram 2000.

Also stored in the memory are a set of consistency rules 14 defining consistency relationships which should be fulfilled by data in the source documents 120; a document

universe definition 140 which defines a selection of the source documents 120 stored in the memory which are to have links corresponding to consistency rules generated; consistency link elements 150 constituting the links within and between the documents; a set 160 of copies of the XML source documents in the document universe definition 140; and an output presentation 170 comprising a set of HTML output documents which can be browsed by a user as will be described in detail below.

The generation of an output presentation 170 will now be described with reference to Figures 8-15. Figure 8 is a flow diagram of the consistency checker 10 of computer apparatus 4 of Figure 1 in accordance with the present invention for the generation of an output presentation.

When the apparatus is initially set up the memory has stored therein the programs and data 20-100 as shown in Figure 5 As a first step for generating an output presentation 170, one or more source document DTD's 110 are generated or imported (s1) and stored in the memory. The source document DTD's can be generated by typing via the keyboard 8 using the XML editor program 20. Alternatively the document DTD's 110 might be imported under the control of the XML editor program 20 into the

memory either via the communications network 5.

After the source document DTD's 100 have been stored in the memory, a number of consistency rules 14 are then generated or imported (s2) and stored in the memory. Consistency rules can be imported and stored in memory via the communications network 5. Alternatively, consistency rules can be generated directly by being input via the keyboard 8 using the consistency rule editor 40 stored in memory as will now be described in detail with reference to Figures 9 and 10.

Figure 9 is a flow diagram of the processing of the consistency rule editor 40 for generating consistency rules. When the consistency rule editor 40 is initially invoked this causes the computer apparatus to show (s22) details of the source document DTD's 100 currently stored in the memory together with details of any consistency rules already stored in memory on the display 7. A user is then prompted to either add, delete or edit a consistency rule 14 from the rules presently stored in memory. When via the keyboard 8 or mouse 9 the computer detects (s23) that a user has input an indication of whether he wishes to add, delete or edit a rule the flow of control is changed accordingly.

If the computer detects that a user has input via the keyboard 8 or mouse 9 that he wishes to add a rule, the user (s24) is prompted to input details of a new consistency rule which is to be added to the consistency rules 14 presently stored in memory as is described in detail with reference to Figure 10.

If the computer detects that a user has input that he wishes to delete a rule a user is prompted (s25) to select one of the rules 14 stored in memory to be deleted and the computer then deletes (s26) the selected rule from the memory.

If the computer detects (s27) that a user has input that he wishes to edit a consistency rule 14 present in the memory the user is prompted to select the rule to be edited (s27). The selected rule is then deleted from memory and the details of the rule are provided as an initial input for the input of a new rule as is described below with reference to Figure 10. The user can then edit (s28) this initial input and the amended rule is then stored in the memory.

After a rule has been added, deleted or edited and the rules 14 stored in the memory updated accordingly, the

user is prompted (s29) whether the rules are now considered complete. If a user indicates via the keyboard 8 or mouse 9 that the rules are considered complete the consistency rule editor 40 stops. If a user indicates that the rules are not yet considered complete, the updated rules 14 now stored in memory are displayed (s21) on the display once more and the user is once again prompted to add, edit or delete a rule.

The following, illustrated by reference to Figure 10, is an example of how the exemplary consistency rule discussed above in relation to Figures 2, 3 and 4 defining in the relationship for a set of software engineering documents produced in UML, that "for every instance in a collaboration diagram there must exist a class in a class diagram of the same name" would be generated by a user using the consistency rule editor 40.

After the computer has detected (s23) that the user wishes to input a new rule, the user is then prompted to input the natural language definition of the rule which was about to be generated. This definition is input (s30) by a user via the keyboard 8 and stored as a natural language definition of the rule. In this example the user would input "for every instance in a

collaboration diagram there must exist a class in a class diagram of the same name".

The user is then prompted to enter a rule type for the new rule. Since, in this example, the rule is a rule which is to mean to hold true for all instances in a collaboration diagram in a set of documents and data within a collaboration diagram document will be inconsistent with the rule if any of the data cannot be linked with other data in a class diagram document, the type of rule CT and this type would therefore entered (s31) as the rule type.

The user then selects (s32) from the displayed DTDs for a UML collaboration diagram a portion of the DTD which would be fulfilled by data representing instances in a collaboration diagram 2000 in source documents for UML diagrams, say for example the portion of the UML DTD for such data is:

root().child(all,Package).(all,Collaboration Diagram). (all,Collaboration).(all,Instance).

This selection is then stored in memory.

The user then selects (s33) from the displayed DTD for a

UML class diagram a portion of the DTD which would be fulfilled by data representing class data in source documents representing a UML class diagram 1000, say for example, this portion of a DTD is defined by:

root().child(all,Package).(all,Class Diagram).(all,Class)

This selection is then also stored in memory.

The user is then prompted to enter (s34) via the keyboard 8 the condition for the rule. In this example, the condition would be that the data found in a source document at the part of a source document to be found as an instance of the first selected portion of a DTD has the attribute class and that this is equal to the value of data found in a source document of the part of a source document to be found as an instance of the second selected portion of a DTD.

A consistency rule is then generated (s35) by the computer in XML comprising a computer generated tag 21, a text description of a rule 22 corresponding to the description input by the user, a rule type 23 corresponding to the input rule type and a pair of X-pointers 24, 25 corresponding to the first and second

portions DTD selected by the user and a set of conditions 26 corresponding to the conditions input by the user.

The rule in this example generated in terms of an XML in accordance with the consistency rule DTD 30 would comprise an XML definition such as:

```
<Consistency Rule        id="rl"        type="CT">
        <id>rl</rl>
        <Description>
        For every instance in a collaboration diagram there must be a class in a class diagram
                                                                    with the same name.
        <Description>
        <Source>
                <XPointer>
                root().child(all,Package).(all,CollaborationDiagram).
                (all,Collaboration).(all,Instance)
                </XPointer>
        </Source>
        <Destination>
                <XPointer>
                root().child(all,Package).(all,ClassDiagram).(all,Class)
                </XPointer>
        </Destination>
        <Condition      expsource="origin().attr(CLASS)"
                        op="equal"
                        expdest="origin().attr(NAME)"/>
        </ConsistencyRule>
```

Although in the above example the condition to be fulfilled by a rule was a simple comparison between two values to check whether they were identical, it will be appreciated that any suitable comparison operation such as >< or ≠ could form part of a condition. It will also be appreciated that more complicated conditions involving the processing of data and comparisons with variables

could instead be used.

After source document DTD's 110 and consistency rules 14 for the document types defined by the DTD's have been stored in the memory, a user is then required to generate or import (s3) source documents corresponding to the document types defined by the DTD's 110 in the memory. Either new documents can be generated using XML editor 20 or by using other application software 15 or already existent XML documents can be down loaded via the communications network 5. All the XML source documents 120 generated or inputted are then stored in the memory.

After the source documents 120 have been stored in the memory a user can then invoke the document universe editor 50. When the document universe editor 50 is invoked, a user then prompted to select from a list of source documents 120 stored in the memory those for which an output presentation 170 is to be generated. When the user has selected all the documents that he wishes to be considered for the generation of consistency links, copies 160 of the XML source documents together with

associated DTD's are stored as the basis for generating an output presentation 170 and a list of those selected documents copied into memory is stored (s4) as a document universe definition 140. The document universe editor 50 then causes the consistency link generator 60 to be invoked.

In the simple example discussed in relation to Figures 2, 3 and 4, the XML source documents downloaded into memory would comprise XML source documents for the business entities class diagram 1000 and collaboration diagram 2000. The document universe editor 50 would enable a user to select both of these source documents and then following the selection of these two documents, copies 160 of the XML source document for business entities class diagram 1000, and the XML source document for the collaboration diagram 2000 together with their associated DTDs would be generated and stored in the memory, together with a document universe definition 140 comprising a list of the copies 160.

After the document universe definition 140 has been

stored in the memory, the consistency link generator 60 is then invoked to generate a set of consistency link elements (s5).

Figure 11 is a flow diagram illustrating the processing of the consistency link generator 60. Initially, the consistency link generator 60 selects the first (s40) rule of the consistency rules 14 stored in memory. A set of source elements (expressed in X-pointer) corresponding to the first rule is then determined (s41) for each of the copies of the XML source documents 160 as listed in the document universe definition 140. This is accomplished by the computer traversing the XML data structure of each XML source document 160 in the document universe definition 140, following the source element X-pointer expression 24 for the rule currently under consideration.

Thus in the case of the exemplary rule and the documents shown in Figure 2 as described above, the consistency link generator 60 initially will use the source element X-pointer 24 for the consistency rule under consideration

to build the set of source elements for the business entities class diagram 1000 and the collaboration diagram 2000, being a set of source elements comprising expressions in X-pointer to the organiser window instance 2001, organiser instance 2002, meeting instance 2003, and date instance 2004 data in the collaboration diagram 2000.

The consistency link generator 60 then generates (s42) a set of destination elements for the rule currently under consideration in a similar manner by traversing the XML data structure of each source document 160 in the document universe definition 160 following the destination element X-pointer expression 25 for the rule currently under consideration.

Thus in the case of the exemplary rule and of the documents shown in Figure 2 as discussed above, the consistency link generator 60 would generate a set of destination elements using the destination element X-pointer expression 25 for the rule under consideration to create a set of destination elements comprising

expressions in X-pointer to the user class 1001, message class 1002, meeting class 1003, location class 1004, date class 1005, organiser class 1006, and participant class 1007 data in the business entities class diagram 1000.

The rule type (s43) of the rule currently under consideration is then determined and links of the appropriate type generated as explained below.

If the rule type is determined to be of type CT, CT consistency link data is generated (s44) as is described in detail with reference to Figures 12 and 13. If the rule type is determined to be of type CF, CF consistency link data is generated (s45) as is described in detail with reference to Figure 14. If the rule type is determined to be of type IF, IF consistency link data is are generated (s46) as is described in detail with reference to Figure 15.

When consistency links have been generated appropriate to the rule currently under consideration and stored as consistency link data 150 the consistency link generator

60 then determines (s47) whether the rule under consideration is the last of the rules stored as a consistency rule 14 in the memory. If it is determined that the rule currently under consideration is not the final rule the next rule (s48) is then considered and new lists of potential source and destination data (s41 & s42) are generated. If it is determined that the rule currently under consideration is the last rule of the consistency rules 14 stored in the memory the consistency rule editor 60 then passes control to the style sheet processor 80 to generate a set of output documents as will be described in detail later.

Figure 12 is a flow diagram illustrating the generation of consistency links corresponding to a rule defining CT type consistency relation. As stated previously a CT type consistency relation is one which is required to be true for all instances of data that exist within the set of documents defined as the document universe definition 140.

After a set of source elements and a set of destination

elements have been generated by the consistency link generator 60 (s41 and s42) and the type of the rule currently under consideration has been determined (s43) to be of a type CT, the consistency link generator 60 then determines (s50) if the set of destination elements is empty. If the destination is determined to be empty the consistency link generator 60 then generates an inconsistent link (s56) as will be described later.

If the set of destination elements is not determined to be empty the consistency link generator 60 then compares (s51) data identified by first element in the set of source elements with data identifying the first element in the set of destination elements. If the data is found to fulfil the requirements of the condition 26 for the rule currently under consideration, a consistent link is then generated (s52) and added to the consistency link elements 150 stored in memory.

Figure 13 is a block diagram of the data structure of a consistency link generated by the consistency link generator 60. A consistency link comprises a locator for

a source document element 200, a locator for a destination document element 210, data 220 identifying the rule currently under consideration which has caused the link to be generated and status data 230 as either consistent or inconsistent.

When the consistency link generator 60 determines that data identified by the element currently under consideration from the set of source elements and data identified by the element currently under consideration from the set of destination elements fulfil the requirements of the condition 26 of the CT type rule currently under consideration, the consistency link generator 60 generates a consistent link in XML comprising copies 200,210 of the elements currently under consideration from the set of source elements and set of destination elements, data 220 identifying the rule currently under consideration and status data 230 indicating that the link was generated because the data linked was consistent with the identified rule.

Thus for example when the consistency link generator 60

determines that for the documents of Figure 2 the Organiser instance 2002 in the collaboration diagram 2000, has corresponding organiser class 1006 in the class diagram 1000 when testing the consistency rule "For every instance in a collaboration diagram there must exist a class in an associated class diagram of the same name", the consistency link generator 60 generates a consistent link comprising a copy 200 of the element data from the set of source elements currently under consideration, being a URL for the XML source document for the collaboration diagram 2000 and an expression in X-pointer which points to the data representing the Organiser instance 2002 in the collaboration diagram 2000; a copy of an element from the set of destination elements, being a URL for the XML source document the business entities diagram 1000, and an expression in X-pointer which points to the data representing the Organiser class 1006 in the business entities class diagram 1000; a copy of the identification tag 21 for the rule currently under consideration as identification data 220; and status data 230 labelling the consistency link as consistent.

After a consistent link (s52) has been generated or it has been determined that the data identified by the element from the set of source elements and the element from the set of destination elements does not fulfil the condition 26 of the rule under consideration, the consistency link generator 60 then determines if the element currently under consideration from the set of destination elements is the last element in the set (s53). If it is determined that the element from the set of destination elements is not the last of the elements in the set of destination elements then the consistency link generator 60 then tests data identified by the element currently under consideration from the set of source elements against the data identified by the next element in the set of destination elements and generates further consistent links if appropriate (s54, S51-S53).

If it is determined that the element currently under consideration from the set of destination elements (s53) is the last in the set of destination elements, the consistency link generator 60 then determines (s55) whether in the course of testing each of the data

identified by the elements in the set of destination elements against the data identified by the element currently under consideration from the set of source elements, any consistent links have been generated. If no links have been generated the consistency link generator 60 then generates an inconsistent link (s56), which is then added to the consistency link data 150 stored in memory.

When the consistency link generator 60 generates an inconsistent link for a CT type rule, link data is generated comprising a copy 200 of the element currently under consideration from the set source elements, a null element 210, data 220 identifying the rule currently under consideration and a status data 230 indicating that the link was generated because no data existed which could be linked as being consistent with the data labelled by the link.

For example, when the consistency link generator 60 determines that for the documents of Figure 2 the Organiser window instance 2001 has not been linked with

any of the class data of the business entity class diagram 1000 the consistency link generator 60 generates an inconsistent link comprising a copy 200 of the element from the source set for the data under consideration, being a URL for the XML source document for the collaboration diagram 2000 and an expression X-pointer which points to the data representing the Organiser window instance 2001 in the collaboration diagram 2000, a null element 210, a copy of the identification tag 21 for the rule currently under consideration as identification data 220 and status data 230 labelling the consistency link as inconsistent.

After an inconsistent link (s56) has been generated or if any consistent links have been created for the data pointed to by the element currently under consideration from the set of source elements, the consistency link generator 60 determines (s57) whether the element currently under consideration from the set of source elements is the last element on the set of source elements. If the element currently under consideration is not the last element in the set of source elements,

4005922 052202

the consistency link generator 60 then compares the next element in the set of source elements with the first element from the set of destination elements (s58, S50-S57).

In this way the consistency link generator 60 ensures that all possible pairings of data identified by elements in the source set and data identified by elements in the destination set are considered for the generation of links and that links are generated whenever data is found which fulfils the condition of the rule under consideration, or data is identified in the source set which does not cause the generation of any consistent links.

Figure 14 is a flow diagram for the generation of consistency links corresponding to a rule defining a CF type consistency relation. As stated previously a CF type consistency relation is one which if data exists which fulfils a consistency rule condition, a consistency link is required to be formed. However, in contrast to a CT type consistency relation, if no data is found which

fulfils the condition of a CF type consistency relation, no inconsistent links are generated.

Since, whenever data is found to exist which fulfils a consistency rule condition for either a CT or CF type consistency rule, a consistent link is generated the processing of data in accordance with a CF type rule where that data is found to fulfil a consistency rule condition is identical to the processing for a CT type consistency rule. The only difference in processing by the consistency link generator 60 between the processing of a CT type consistency rule a CF type consistency rule therefore arises, if no consistent links are generated for a piece of data identified by an element from the set of source elements.

The flow diagram of Figure 14 is therefore identical to the flow diagram of Figure 12 for the generation of links according to a CT type consistency rule except that in the case of a CF consistency rule when the consistency link generator 60 determines (s50) that no elements are present in the set of destination elements or no

consistent links have been generated for data identified by a source element under consideration the consistency link generator 60 does not then generate an inconsistent link, rather the consistency link generator omits this step (s55,s56) and proceeds to determine (s57)whether the element under consideration from the source set is the last element in the source set and then proceeds either to compare that data identified by the next element in the source set with data identified by the destination set (s58, s50-s54) or stops.

By following a flow of control identical to that for the processing of CT type rules but omitting the step (s55-s56) of generating inconsistent links when no data is found not to fulfil the condition 26 of the rule under consideration with data identified by an element in the set of source elements, the consistency link generator 60 generates links in accordance with a consistency relationship which it is not mandatory for data in documents to fulfil.

Figure 15 is a flow diagram illustrating the generation

of consistency links corresponding to a rule defining an IF type consistency relation. As previously stated an IF type consistency relation is one which is required not to be true for any instances of data within the set of documents defined as the document universe definition 140.

The processing of an IF type consistency rule is identical to a CT type consistency rule except that when data is identified which fulfils the conditions 26 of rule under consideration a link to the data which fulfils the condition 26 is generated and labelled inconsistent. If no data identified by the elements in the set of destination elements which fulfils the consistency condition 26 of the rule under consideration with data identified by an element under consideration from the source set, a link is made the data identified by the element under consideration from the set of source elements, labelling the data as consistent.

Figure 15 is therefore identical to Figure 12 except that the status of links generated in the steps when

generating a consistent (s52) or inconsistent (s54) link are interchanged.

Thus inconsistent links comprising the pair of elements 200,210 under consideration from the sets of source elements and destination elements, data 220 identifying the rule under consideration, and status data 230, indicating a link was generated because data linked was inconsistent with an identified rule are generated (s60) when the data identified by the elements under consideration does fulfil the condition 26 of the rule under consideration (s51).

And, if the element under consideration is the last element in the set of destination elements and no links have already been generated for data identified by the element currently under consideration from the source set (s55) then the consistent link generator 60 generates a consistent link with status data 230, labelling the link as consistent (s61).

When the consistency link generator 60 has generated

consistency link elements for all the consistency rules 14 stored in memory that relate to the source documents 160 and consistency rules 14 all the data relevant to the fulfilment of required consistency relationships is available and represented in XML.

The source documents 160, consistency rules 14, and the consistency link elements 150 are therefore passed to the style sheet processor 80 which processes them in accordance with the XSL output presentation style sheet 70 to generate (s6) an output presentation 170. The style sheet processor 80 could be any of the style sheet processors known in the art such as Microsoft MSXSL with the presentation style sheet merely identifying how the data from the source documents 160, consistency rules 14 and consistency link elements 150 are to be rendered as is well known in the art.

Thus for example, in the case of the output presentation of Figure 3, the style sheet 70 is such as to interpret all the consistency link elements 150 as the basis for generating links 3001-3003 to the appropriate parts of

HTML documents corresponding to the data of the 160 XML source documents for the UML diagrams 1000,2000. The style sheet 70 is also arranged to include in the HTML presentation labels 3004,3005 comprising the natural language description 22 for the rule identified as a rule which causes consistency link elements to be generated.

When the output presentation 170 has been generated, in accordance with the present embodiment of the present invention, the output presentation 170 generated by the style sheet processor 80 comprises an HTML file that can be viewed (s7) with a standard browser such as Netscape Navigator or Microsoft Internet Explorer.

After an output presentation has been generated for a set of consistency rules 14 and a document universe 140 it is necessary to update the output presentation whenever significant changes are made to the XML source documents referenced in the document universe definition. In this embodiment the maintenance of the correspondence of an output presentation 170 with a set of XML source documents is achieved by the monitor module 90 and the

consistency manager 100 as will now be described.

The processing of monitor module 90 and the consistency manager 100 for monitoring and maintaining an output presentation 170 corresponding to a set of source documents 120, is illustrated by Figure 16.

After the generation of an output presentation 170, the monitor module 90 is arranged to monitor whether the source documents 120 stored in memory have been updated (s70). In a simple embodiment of the monitor module 90 this could be achieved by periodically checking whether the time stamp for the time of storage of a document 120 in the memory has varied, which would indicate that the previous copy of that source document had been replaced by a newer copy.

If the monitor module 90 detects that an update has occurred the monitor module 90 then determines (s71) which parts of the source document have been changed. One way in which the monitor module 90 could establish which parts of a source document have been changed would

be by performing a difference operation between the source document 120 and the copy of the old source document 160 stored for the purposes of generating the output presentation 170. A suitable difference operation would be a tree difference operation such as the XML tree diff library from IBM Alpha Works.

If the updated version of the source document 120 monitored by the monitor module 90 is determined to be identical to that previously stored the difference operation will indicate that no change has occurred and no further action is undertaken. If however, the monitor module 90 does detect that a change has occurred, the monitor module 90 then invokes the consistency manager, to update the output presentation 170 stored in memory as will now be described.

When the consistency manager 100 is first invoked, the consistency manager 100, first copies the latest version of the updated source document to replace the previous version of that document.

The consistency manager 100 then compares (s72) the portion of the source document 120 which has been amended with the source element expressed in X-pointer 24 and destination element expressed in X-pointer 25 of each of the consistency rules 14 stored in memory to establish which of the consistency rules 14 relates if any to data which may have been affected by the update.

If the update detected by the monitor module 90 is determined to have potentially affect data for which consistency links may have been generated, the consistency manager 100 then (s73) invokes the consistency link generator 60 to update the consistency link data 150 in memory prior to invoking the style sheet processor 80 to generate (s74) output presentation 170 on the basis of the updated copies of source documents 160 and new set of consistency link elements. If no links are determined to be affected by the update, the consistency manager 100 stops.

In this way the checking of a set of source documents against a set of consistency rules is incremental and the

output presentation 170 is made to reflect any changes made to the source documents. It is therefore possible to ensure that source documents fulfil required consistency relationships at key points within the production process.

Figure 17 is a flow diagram showing in detail how the consistency manager 100 determines which consistency rules are affected by an update and instructs the consistency link generator 60 to generate an appropriate updated set of links.

When the consistency manager 100 is first invoked the consistency manager compares the data indicating the portion of the source document 120 which has been updated with the consistency link data 150. The consistency manager 100 then deletes (s80) any of the affected consistency links in the consistency link element 150. The consistency manager 100 then compares (s81) the data representing the updated portion of the document under consideration with the source element 24 of each of the consistency rules 14 stored in memory in turn. If the

consistency manager 100 determines that the source element 24, of a consistency rule under consideration is such that it may refer to data which has been updated, the consistency rule under consideration is determined to be one which may have been affected by the amendment to the source document. The consistency manager 100 then generates (s82) a set of source elements for the data within the updated portion of the updated document in a manner similar to the way the consistency link generator 60 determines a set of source elements for all documents in the document universe 140. The consistency manager 100 then generates (s83) a set of destination elements for all the documents in the document universe in a similar manner to the generation of a set of destination elements by the consistency link generator 60. The consistency manager program 100 then invokes the consistency link generator 60 to generate (s84) a set of consistency links by comparing data identified by elements in the source set and destination set in accordance with the rule under consideration. These consistency links are added to the consistency link element 150 stored in memory.

US006772252B2 20040803

After the consistency link generator 100 has generated new consistency links related to the portion of the source document which has been updated, the consistency manager 100 then determines (s85) whether the set of destination elements 25 of the rule currently under consideration may refer to data which has been updated in the source document. If the consistency manager 100 determines that the set of destination elements 25 of the consistency rule currently under consideration is such that it may refer to data which has been updated in a source document, the consistency manager 100 then generates (s86) a set of source elements for all of the other documents in the document universe 140 in a similar manner to the consistency link generator 60. The consistency manager 100 then determines a set of destination elements S87 for data within the portion of the source document which has been updated and which is identified by the destination element of the consistency rule under consideration. The consistency manager then invokes (s88) the consistency link generator 60 to generate a set of consistency links based upon a comparison identifying elements form the source set and

FOOS7825.052202

elements in the destination set which are then added to the consistency link element 150 stored in memory.

After any further consistency links have been generated, the consistency manager then determines (s89) whether the rule currently under consideration is the last of the rules 14 stored in memory. If the rule currently under consideration is not the last of the rules stored in memory the consistency manager 100 then repeats the process of deleting and regenerating links for the next (s90, s80-s88) consistency rule stored in memory.

In this way the consistency manager deletes any consistency links which may be affected by the update of a portion of a document and then generates new consistency links for the updated portion of the document for all of the consistency rules. Then the consistency manager 100 invokes the style sheet processor 80 to generate an updated output presentation 170 to account for the changes to the XML source documents 160 and reflect the changes made to the consistency links 150.

2025220052 07/22/2025

Although a description of a distributed computer apparatus embodying the present invention has been made from which a single consistency checker has been provided it would be appreciated that one or more consistency checkers could be provided in a computer network each of which could generate their own output presentations. It would be also appreciated that consistency checkers could be provided within a server of a distributed computer network rather than being stored within a terminal. It will also be appreciated that a number of different monitor programs could be provided each of said monitor programs monitoring the update of source documents in the different computer terminals. It would also be appreciated that instead of being provided on a computer network, the consistency checker of the present invention could be provided within a single computer.

Although the previous embodiment of the present invention has been described in terms of a consistency checker in which three types of rules are supported, each of the rules involving the comparison of two pieces of data, it would be appreciated that other types of rules could also

exist. In particular, it would be appreciated that instead of a comparison based upon a pair of data identified by pointers, rules based on comparisons of one or more pieces of data within structured documents could be supported. It would also be appreciated that the testing of more complex rules could involve the processing of those rules in ways other than simple comparisons for example comparisons based upon the use of variables.

Although the embodiments of the invention described with reference to the drawings comprise computer apparatus and processes performed in computer apparatus, the invention also extends to computer programs, particularly computer programs on or in a carrier, adapted for putting the invention into practice. The program may be in the form of source or object code or in any other form suitable for use in the implementation of the processes according to the invention. The carrier be any entity or device capable of carrying the program.

For example, the carrier may comprise a storage medium,

WO 2005/122026 PCT/GB2005/000741

such as a ROM, for example a CD ROM or a semiconductor ROM, or a magnetic recording medium, for example a floppy disc or hard disk. Further, the carrier may be a transmissible carrier such as an electrical or optical signal which may be conveyed via electrical or optical cable or by radio or other means.

When the program is embodied in a signal which may be conveyed directly by a cable or other device or means, the carrier may be constituted by such cable or other device or means.

Alternatively, the carrier may be an integrated circuit in which the program is embedded, the integrated circuit being adapted for performing, or for use in the performance of, the relevant processes.

1006P2-0520

CLAIMS

1. A method of generating consistency link data within a computer apparatus, said consistency link data, identifying data within structured documents which does or does not fulfil the requirements of a formal set of consistency relationships, comprising the steps of:

inputting a set of one or more structured documents containing data;

storing a set of one or more consistency rules, each of said consistency rules defining a relationship which may be fulfilled by data within structured documents having pre-defined structures; and

processing said structured documents and said consistency rules to generate said consistency link data, identifying within said set of structured documents, data which does or does not fulfil said relationships defined by said set of consistency rules.

2. A method in accordance with claim 1, wherein said generation of consistency link data comprises generating consistency link data identifying data within said set of structured documents data which does fulfil said relationships defined by said set of consistency rules.

3. A method in accordance with claim 1 or claim 2, wherein said generation of consistency link data comprises generating consistency link data identifying data within said set of structured documents data which does not fulfil said relationships defined by said set of

consistency rules.

4. A method in accordance with any of claims 1 to 3, further comprising the step of selecting one or more documents from said input set of structured documents containing data, wherein said step of processing said structured documents and said consistency rules comprises processing said selected documents to generate consistency link data identifying data within said selected documents which does or does not fulfil said relationships defined by said set of consistency rules.

5. A method according to any preceding claim, wherein said step of processing said structured documents and said consistency rules to generate consistency link data comprises, for rules in said set of consistency rules, the steps of:

identifying within said structured documents, data which may fulfil the relationship defined by said consistency rule;

testing said identified data to determine which of said data fulfils or does not fulfil the relationship defined by said consistency rule; and

generating consistency link data, identifying data which does or does not fulfil a required consistency relationship on the basis of said determination.

6. A method according to claim 5, wherein each of said consistency rules in said set of consistency rules is associated with one or more pointers, said pointers being

pointers to parts of a predefined structure for structured documents containing data, wherein said identification step comprises identifying data within documents structured in accordance with said predefined structure, said data being within said portion of said document identified by said one or more pointers.

7. A method according to claim 5 or 6, wherein each of said consistency rules is associated with a condition wherein said testing of identified data comprises determining whether said identified data does or does not fulfil the condition associated with said consistency rule.

8. A method according to claim 7, wherein said condition associated with each of said consistency rules comprises a comparison operation comparing the data identified by said identification step.

9. A method in accordance with claim 7 or 8, wherein each of said consistency rules is associated with a first or a second data generation category, wherein where a said consistency rule is associated with said first data generation category, if it is determined that for data identified which may fulfil a relationship defined by said consistency rule, no other data exists which fulfils with said data said condition associated with said consistency rule, consistency link data is generated identifying said data for which no other data which fulfils said condition and where a said consistency rule

is associated with said second data generation category, if it is determined that for data identified which may fulfil a relationship defined by said consistency rule no other data exists which fulfils with said data said condition associated with said consistency rule no consistency link data identifying said data is generated.

10. A method according any of claims 7 to 9, wherein each of said consistency rules is associated with a link status category, wherein said generation of consistency link data further comprises the generation of data identifying the consistency rule which causes the generation of said consistency link data and status data indicative of whether said identified data does or does not fulfil the relationship defined by said consistency rule, wherein the value of said status data is determined by whether identified data does or does not fulfil the condition associated with said consistency rule and the link status category associated with said consistency rule.

11. A method in accordance with any preceding claim further comprising the step of:

inputting for storage a set of one or more consistency rules, each of said consistency rules defining a relationship which may be fulfilled by data within structured documents having pre-defined structures.

12. A method in accordance with any preceding claim

further comprising the steps of:

monitoring said set of one or more structured documents containing data to determine whether any of said documents has been amended; and

initiating the processing of said structured documents and said consistency rules to generate further consistency link data on the basis of said determination.

13. A method in accordance with claim 12 wherein said monitoring step comprises determining whether the time stamp of a document in a set of one or more structured documents has been amended and if the time stamp of a document has been amended, determining which if any portions of data within said document have been updated.

14. A method in accordance with claim 13, wherein the generation of further consistency link data comprises the steps of:

determining which consistency link data of previously generated consistency link data may be affected by a change made to said identified portions of said structured documents;

deleting any of said previously generated consistency link data which may have been affected by said amendment of said identified portions; and

generating further consistency link data comprising consistency link data identifying data within said identified portions of said documents.

15. A method of generating an output presentation

identifying data within structured documents which does or does not fulfil the requirements of a formal set of consistency relationships defined by a set of consistency rules comprising the steps of:

generating consistency link data in accordance with any preceding claim;

processing said consistency link data to generate an output presentation; and

outputting said output presentation.

16. A method according to claim 15, wherein said processing said consistency link data comprises processing said consistency link data and said set of one or more structured documents to generate an output presentation comprising a set of linked documents, said set of linked documents comprising a set of one or more documents, containing data being representations of said structured documents, wherein the data of said linked set of documents is linked on the basis of said consistency link data thereby identifying data which does or does not fulfil a set of formal consistency relationships defined by a set of consistency rules.

17. A method in accordance with claim 16, wherein said processing said consistency link data further comprises processing said set of one or more consistency rules defining said formal set of consistency requirements wherein each of said consistency rules is associated with label data and said output presentation further comprises labels labelling data identified by said consistency link

data, as fulfilling or not fulfilling the requirements of a consistency relationship defined by a said consistency rule.

18. A method in accordance with claim 17, wherein each of said consistency rules is associated with a natural language description of the relationship which should be fulfilled by data in accordance with the consistency relationship defined by said consistency rule, wherein said labels in said output presentation comprise said natural language description.

19. A computer apparatus for generating consistency link data, said consistency link data, identifying data within structured documents which does or does not fulfil the requirements of a formal set of consistency relationships, comprising:

means for inputting a set of one or more structured documents containing data;

means for storing a set of one or more consistency rules, each of said consistency rules defining a relationship which may be fulfilled by data within structured documents having pre-defined structures; and

means for processing said structured documents and said consistency rules to generate said consistency link data, identifying within said set of structured documents, data which does or does not fulfil said relationships defined by said set of consistency rules.

20. An apparatus in accordance with claim 19, wherein

said means for processing said structured documents and said consistency rules is arranged to generate consistency link data identifying data with in a set of structured documents, data which does fulfil said relationships defined by a said set of consistency rules.

21. An apparatus in accordance with claim 19 or 20, wherein said means for processing said structured documents and said consistency rules is arranged to generate consistency link data identifying data with in a set of structured documents, data which does not fulfil said relationships defined by a said set of consistency rules.

22. An apparatus in accordance with any of claims 19 to 21, further comprising selection means for selecting one or more documents from an input set of structured documents containing data, wherein said means for processing said structured documents and said consistency rules is arranged to generate consistency link data identifying data within said documents selected by said selection means which does or does not fulfil said relationships defined by said set of consistency rules.

23. An apparatus according to any of claims 19 to 22, wherein said means for processing said structured documents and said consistency rules to generate consistency link data comprises:

means for identifying within said structured documents data which may fulfil the relationship defined

by a said consistency rule;

means for testing said identified data to determine which of said data fulfils or does not fulfil the relationship defined by a said consistency rule; and

means for generating consistency link data, identifying data which does or does not fulfil a required consistency relationship on the basis of said determination.

24. An apparatus according to claim 23, wherein said means for storing a set of one or more consistency rules is arranged for storing a set of consistency rules, wherein each of said consistency rules in said set of consistency rules is associated with one or more pointers, said pointers being pointers to parts of a predefined structure for structured documents containing data, and wherein said means for identifying is arranged to identify data utilising a set of pointers associated with a said consistency rule.

25. Apparatus according to claim 23 or claim 24, wherein said means for storing said one or more consistency rules is arranged to store consistency rules when each of said consistency rules is associated with a condition, wherein said means for testing said identified data is arranged to test data by determining whether identified data does or does not fulfil a condition associated with a said consistency rule.

26. An apparatus according to claim 25, wherein said

means for testing is arranged to test identified data on the basis of a comparison operation associated with each of said consistency rules.

27. An apparatus according to claim 25 or claim 26, wherein said means for storing a set of one or more consistency rules is arranged to store a set of consistency rules, each of said consistency rules having associated therewith a first or second data generation category, wherein said means for generating consistency link data is arranged to generate consistency link data where said consistency rule is associated with a first data generation category and it is determined that for data identified which may fulfil a relationship defined by said consistency rule no other data exists which fulfils with said data said condition associated with said consistency rule and wherein said means for generating consistency link data is arranged not to generate consistency link data where a said consistency rule is associated with said second data generation category and it is determined that the data identified which may fulfil that relationship defined by said consistency rule no other data exists which fulfils with said data said condition associated with said consistency rule.

28. An apparatus according to any of claims 25 to 27, wherein said means for storing a set of one or more consistency rules, is arranged to store a set of consistency rules, each of said consistency rules having

an associated link status category, wherein said means for generating consistency link data is arranged to generate data identifying the consistency rule which causes the generation of said consistency link data and status data indicative of whether said identified data does or does not fulfil the relationship defined by said consistency rule, wherein the value of said status data is determined by whether identified data does or does not fulfil the condition associated with said consistency rule and the link status category associated with said consistency rule.

29. An apparatus in accordance with any of claims 19 to 28 further comprising,

means for inputting a set of one or more consistency rules for storage in said means for storing.

30. An apparatus in accordance with any of claims 19 to 29, further comprising:

monitoring means for monitoring a set of one or more structured documents containing data to determine whether any of said documents have been amended; and

initiation means for initiating the generation of further consistency link data on the basis of said determination.

31. An apparatus in accordance with claim 30, wherein said monitoring means is arranged to determine whether the time stamp of a document in said set of one or more structured documents has been amended and if the time

stamp of a document has been amended, to determine which if any portions of data within said document have been updated.

32. An apparatus in accordance with claim 31, further comprising determination means for determining which consistency link data of previously generated consistency link data may be affected by change made to identified portions of a said structured document; and

deletion means for deleting any of said previously generated consistency link data which may have been affected by said amendment of said identified portions, wherein said means for processing said structured documents and said consistency rules to generate said consistency link data is arranged to generate further consistency link data comprising consistency link data identifying data within said identifying portions of said documents following said deletion by said deletion means.

33. A computer apparatus for generating an output presentation identifying data within structured documents which does or does not fulfil the requirements of a set of consistency relationships defined by a set of consistency rules comprising:

computer apparatus for generating consistency link data in accordance with any of claims 19 to 32,

processing means for processing said consistency link data to generate an output presentation; and

output means for outputting said output presentation.

34. An apparatus in accordance with claim 33, wherein said processing means is arranged to process said consistency link data and said set of one or more structured documents to generate an output presentation comprising a set of link documents, said set of linked documents comprising a set of one or more documents containing data being representations of said structured documents, wherein the data of said linked set of documents is linked on the basis of said consistency link data thereby identifying data which does or does not fulfil a set of formal consistency relationships defined by said set of consistency rules.

35. An apparatus in accordance with claim 33 or 34, wherein said processing means is arranged to process said set of one or more consistency rules defining said formal set of consistency requirements wherein said means for storing a set of one or more consistency rules is arranged to store a set of consistency rules, each of said consistency rules having associated therewith label data, said processing means being arranged to generate an output presentation further comprising labels labelling data identified by said consistency link data as fulfilling or not fulfilling the requirements of a consistency relationship defined by a set of consistency rules.

36. An apparatus in accordance with claim 35, wherein said means for storing a set of one or more consistency rules is arranged to store a set of one or more

consistency rules each having associated therewith a natural language description of the relationship which should be fulfilled by data in accordance with the consistency relationship defined by said consistency rule, wherein said processing means is arranged to generate an output presentation wherein said natural language descriptions are used to label data which does or does not fulfil a requirement of a set of consistency rules.

37. A computer apparatus for generating a set of consistency rules for use in any of the computer apparatus of claims 19 to 32, said rules defining relationships which may be fulfilled by data with in structured documents having a predefined structure, said apparatus comprising:

selection means for selecting one or more portions of a data structure defining the structure of a document which is to be filled by data; and

input means for inputting a condition against which data from within documents in accordance with said predefined structure found at said portions is to be tested to determine whether date does or does not fulfil a required relationship.

38. An apparatus in accordance with claim 37, wherein said input means is further arranged for inputting a data generation category to be associated with said consistency rule, wherein said data generation categories determine whether or not consistency link data is

generated for said rule when no data exists which fulfils the condition of said consistency rule.

39. An apparatus in accordance with claim 37 or claim 38, wherein said input means is further arranged for inputting a links data category, wherein said category such as to determine the value of status data associated with consistency link data generated in accordance with said consistency rule in dependence upon whether date identified does or does not form the condition the associated with said consistency rule.

40. An apparatus in accordance with any of claims 37 to 39, wherein said input means is further arranged for inputting a natural language description of a consistency relationship defined by a said consistency rule, wherein said natural language description is arranged to be used for the generation of labels in an output presentation.

41. A data carrier having recorded thereon a computer program for generating within a computer an apparatus in accordance with any of claims 19 to 40.

42. A data carrier in accordance with claim 41 comprising a computer disc.

43. A computer disc in accordance with claim 42, wherein said computer disc comprises an optical magneto-optical or magnetic disc.

SUBSTITUTE SHEET (RULE 26)

44. A data carrier in accordance with claim 41, comprising an electrical signal transferred via the Internet.

45. A method for generating consistency link data within a computer apparatus, said consistency link data, identifying data within structured documents which does or does not fulfil the requirements of a formal set of consistency relationships, substantially as herein described with reference to any of the accompanying drawings.

46. An apparatus for generating consistency link data, said consistency link data, identifying data within structured documents which does or does not fulfil the requirements of a formal set of consistency relationships, substantially as herein described with reference to any of the accompanying drawings.

WO 99/22322 PCT/US98/22822

ABSTRACT

METHOD AND APPARATUS FOR MONITORING AND MAINTAINING THE CONSISTENCY OF DISTRIBUTED DOCUMENTS

A computer network comprising a plurality of terminals (1-4) connected via a communications network 5 is provided. In each of the terminals (1-4) application software (15) for generating XML source documents is provided. XML source documents created using the application software (15) are then sent to one of the terminals (4) having stored therein a consistency checker (10). The consistency checker (10) processes the source document together with a set of consistency rules (14) to generate an output presentation identifying data within the source documents which does or does not fulfil the requirements of consistency relationships defined by the consistency rules (14).

Refers to Figure 1

10009722 052202

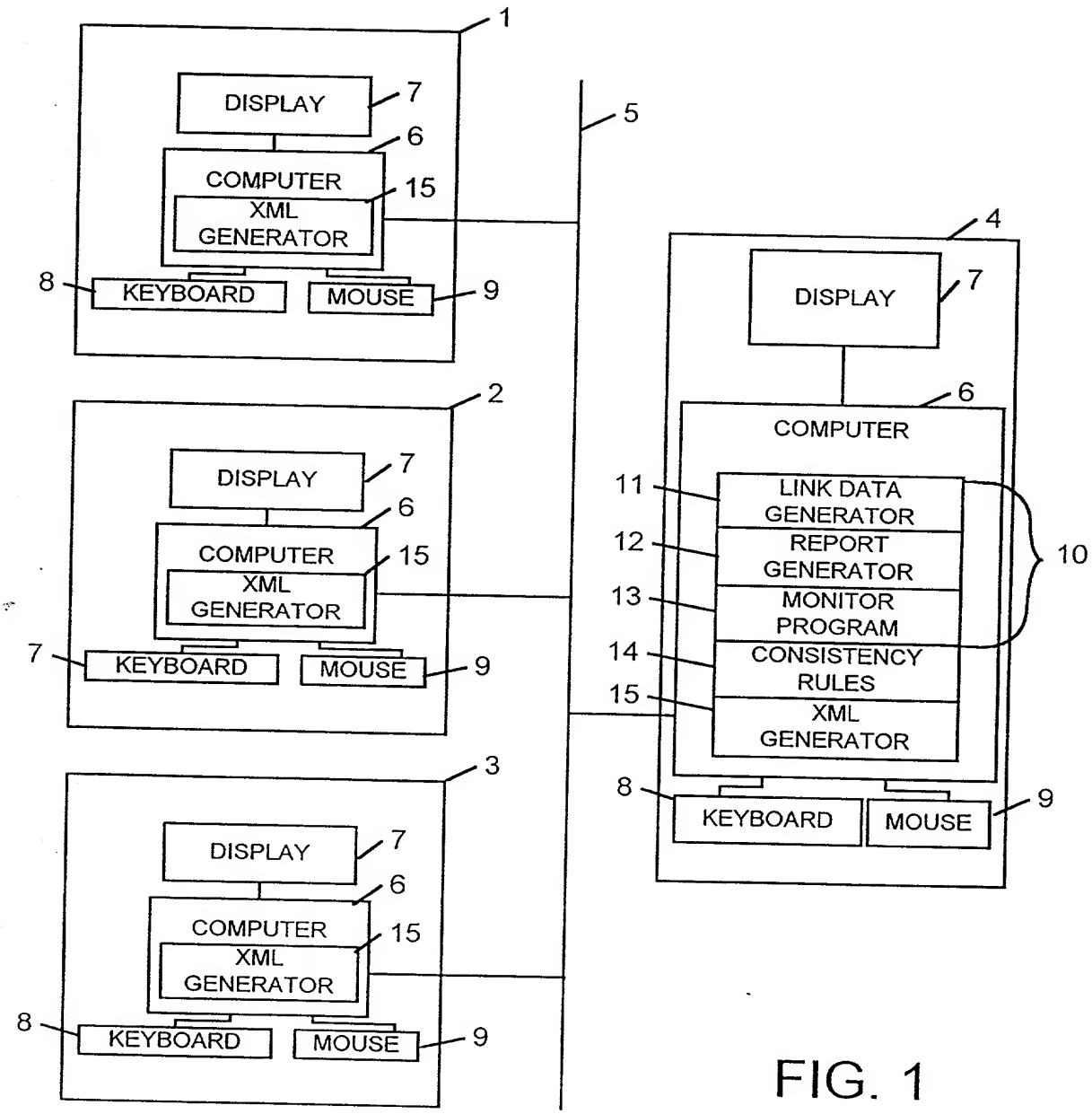
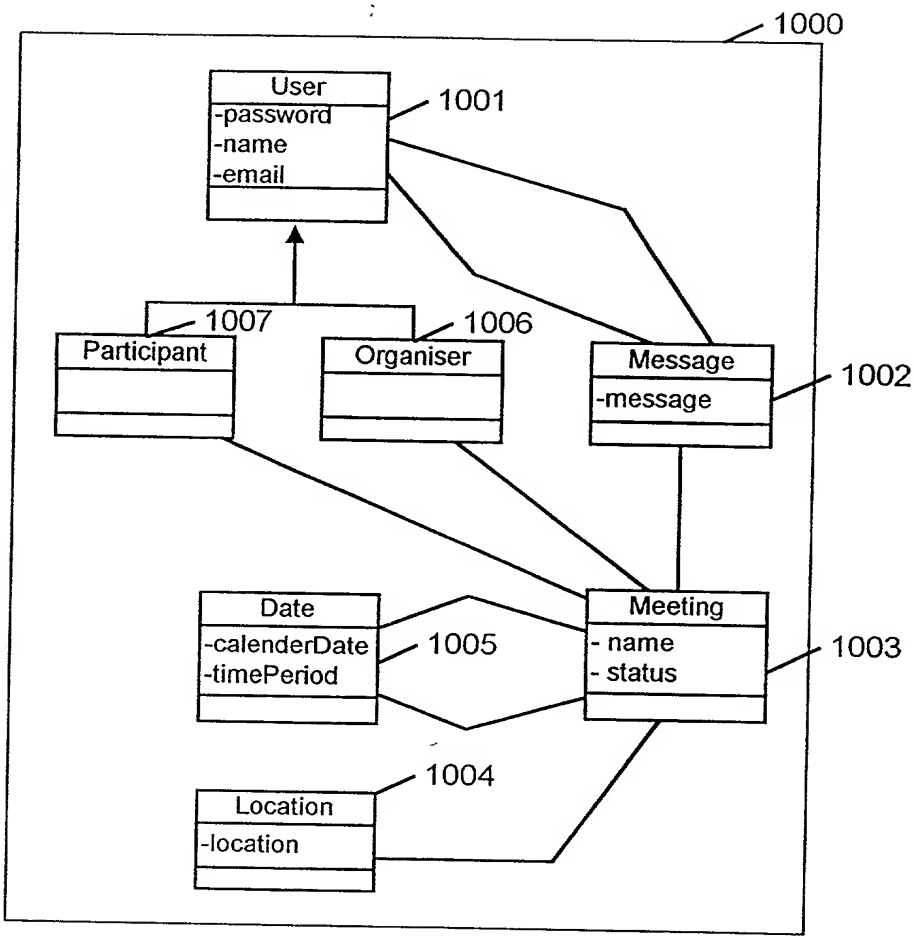


FIG. 1



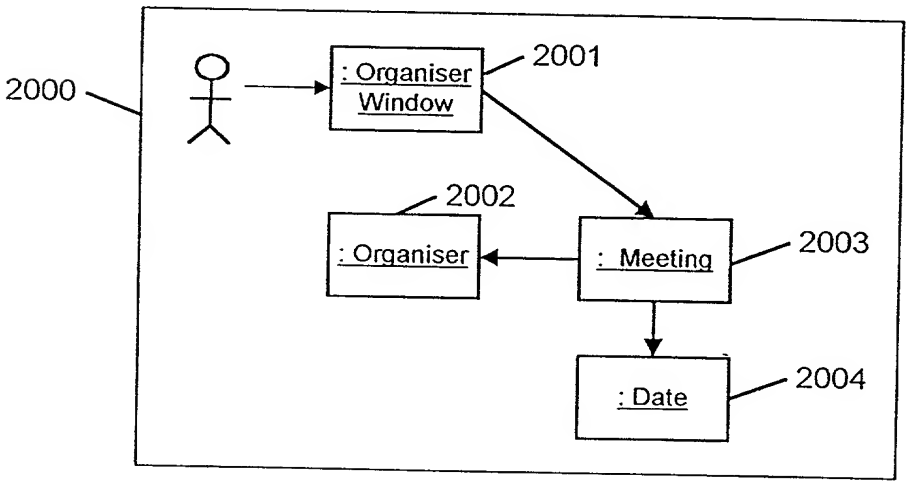


FIG. 2

10009722 052202

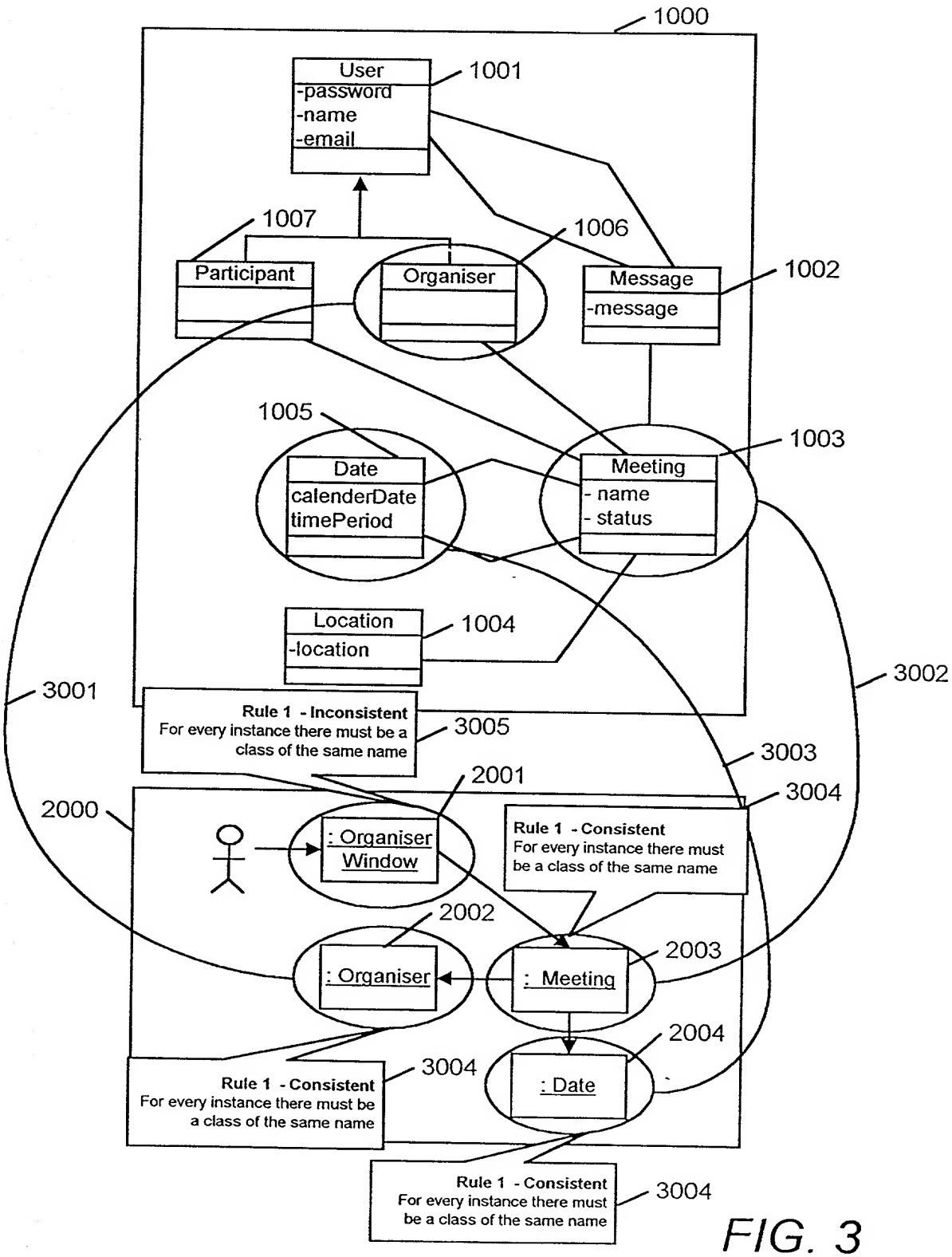


FIG. 3

10009722 - 050202

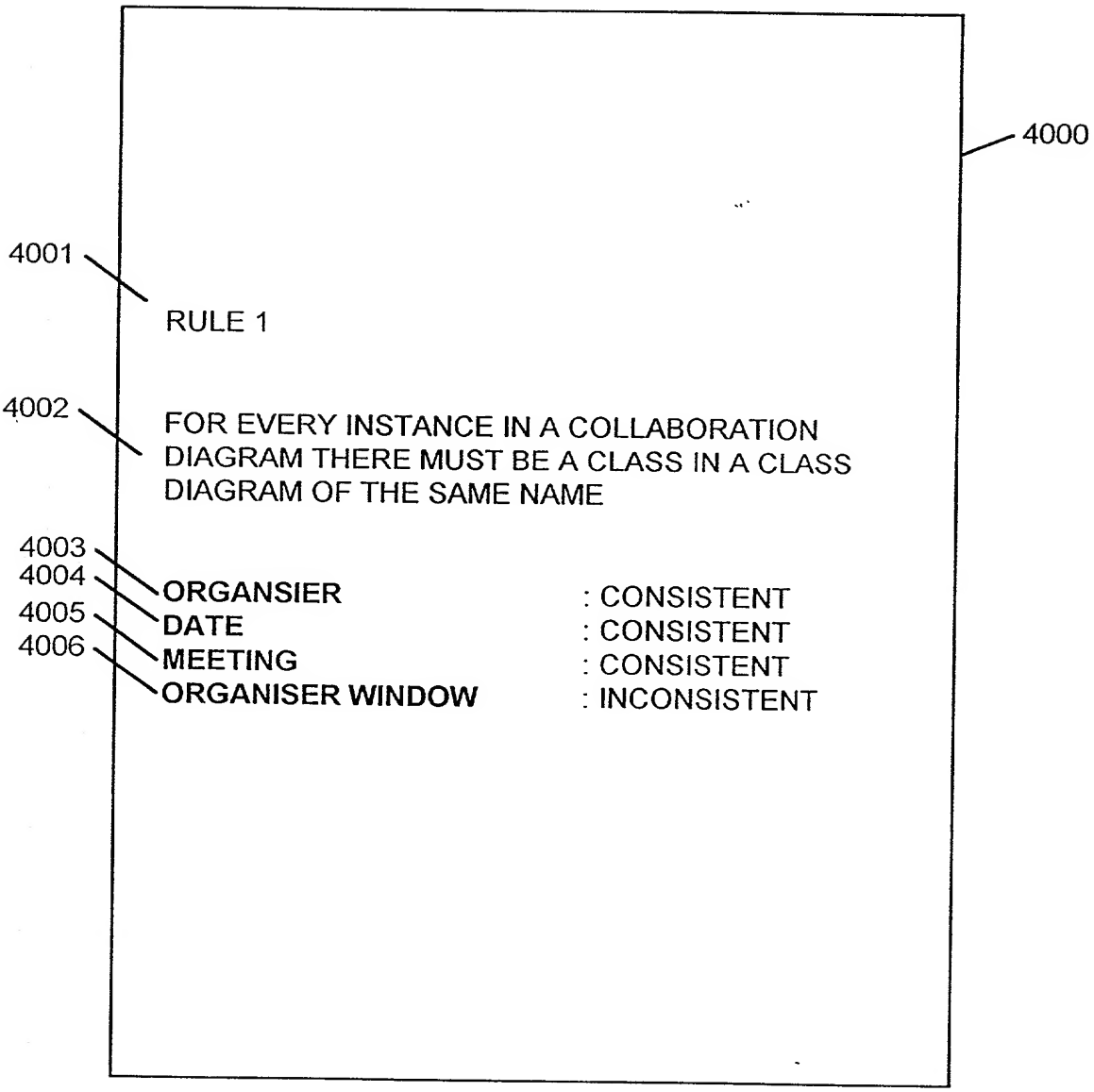


FIG. 4

10009722 022202

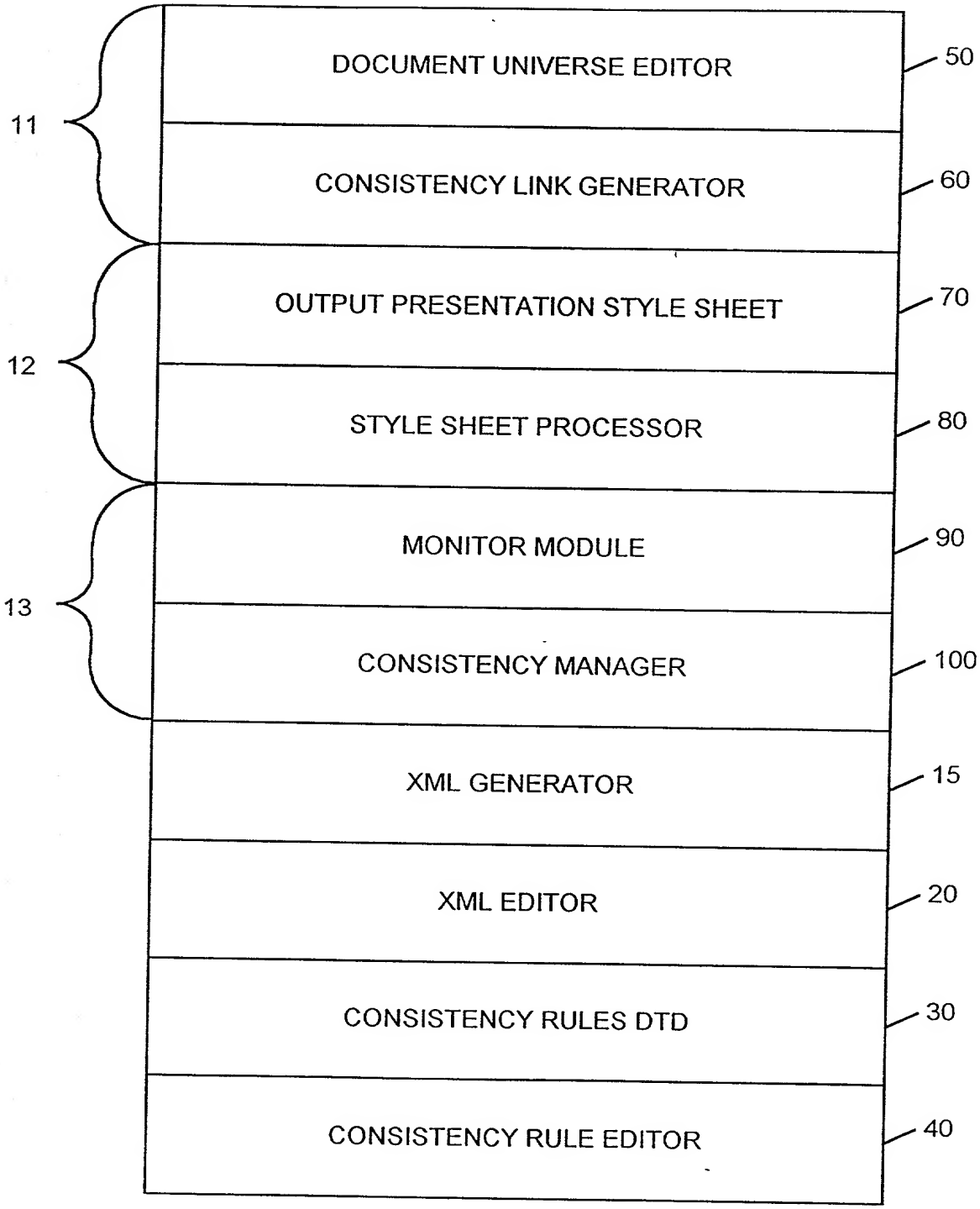


FIG. 5

10009722 052202

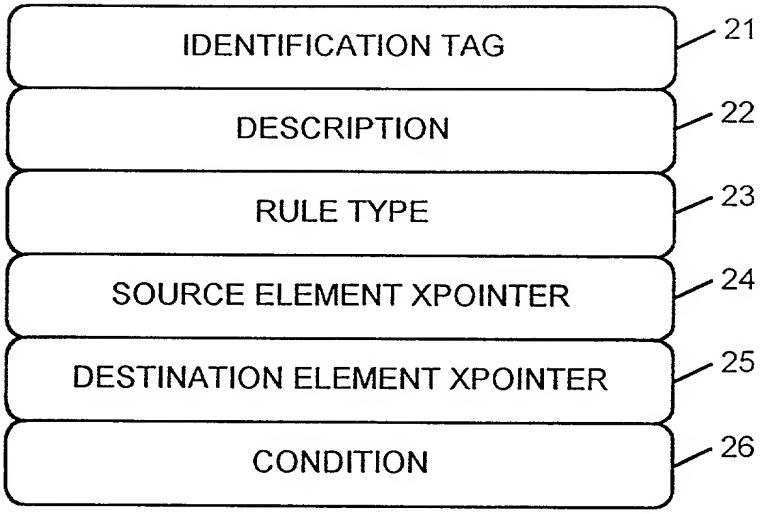


FIG. 6

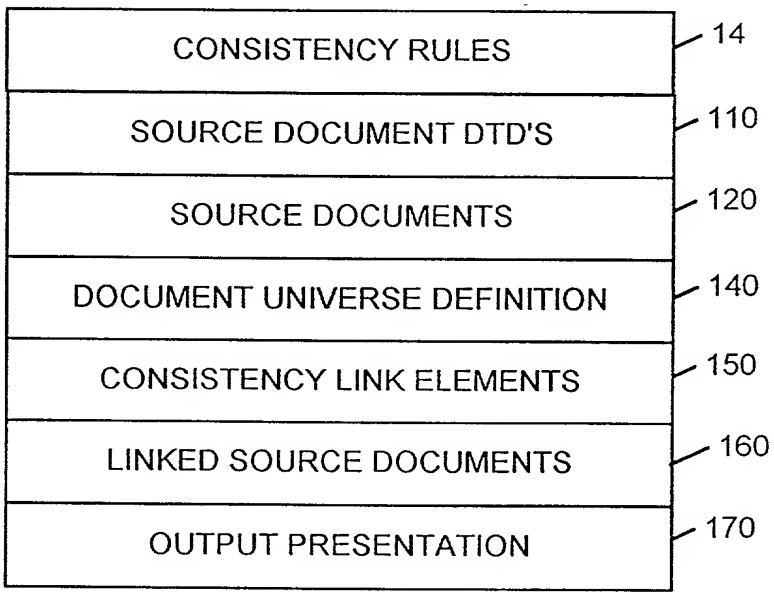


FIG. 7

10009722.052202

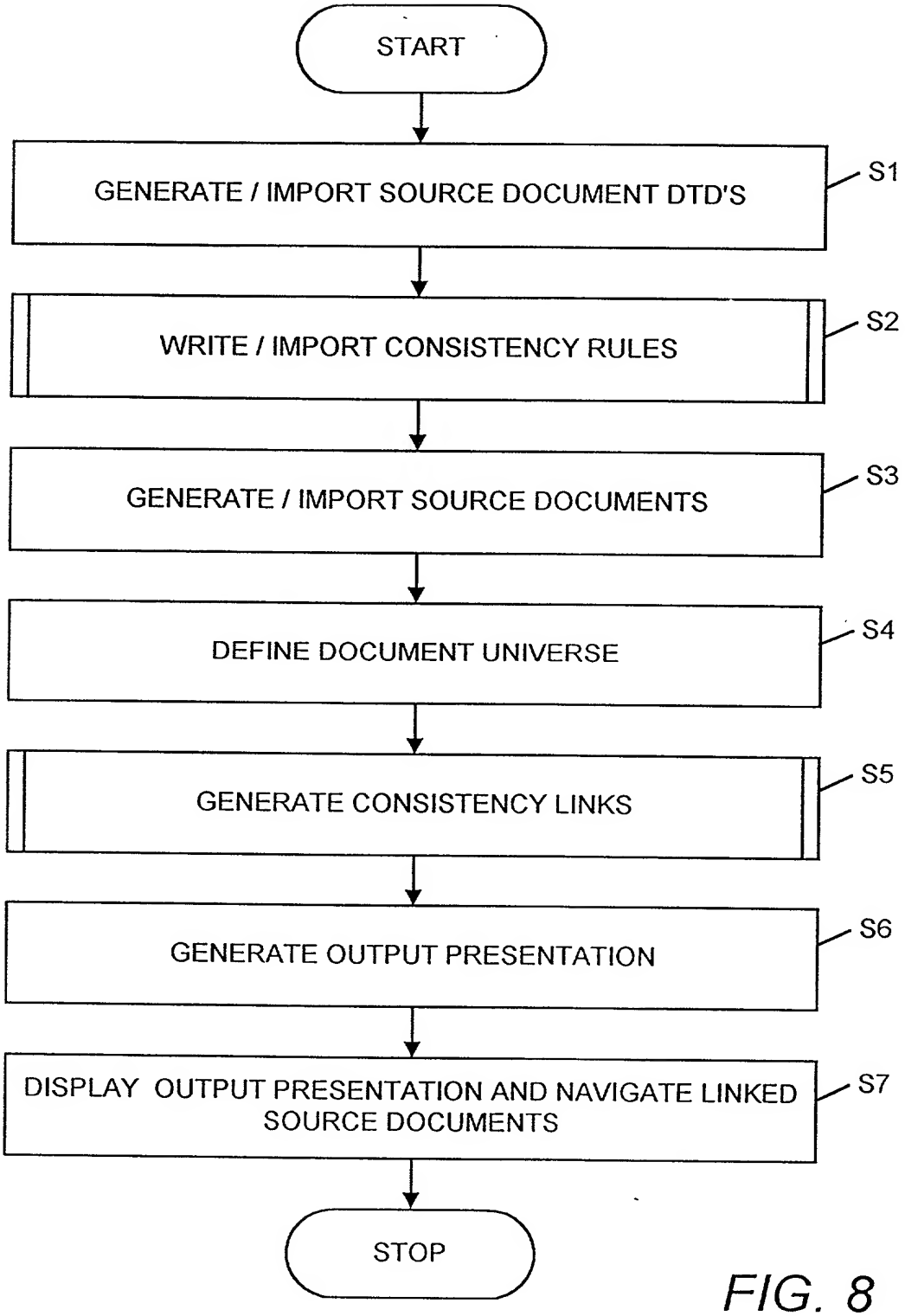


FIG. 8

10009722 020202

10/009722 25 Sep 2002

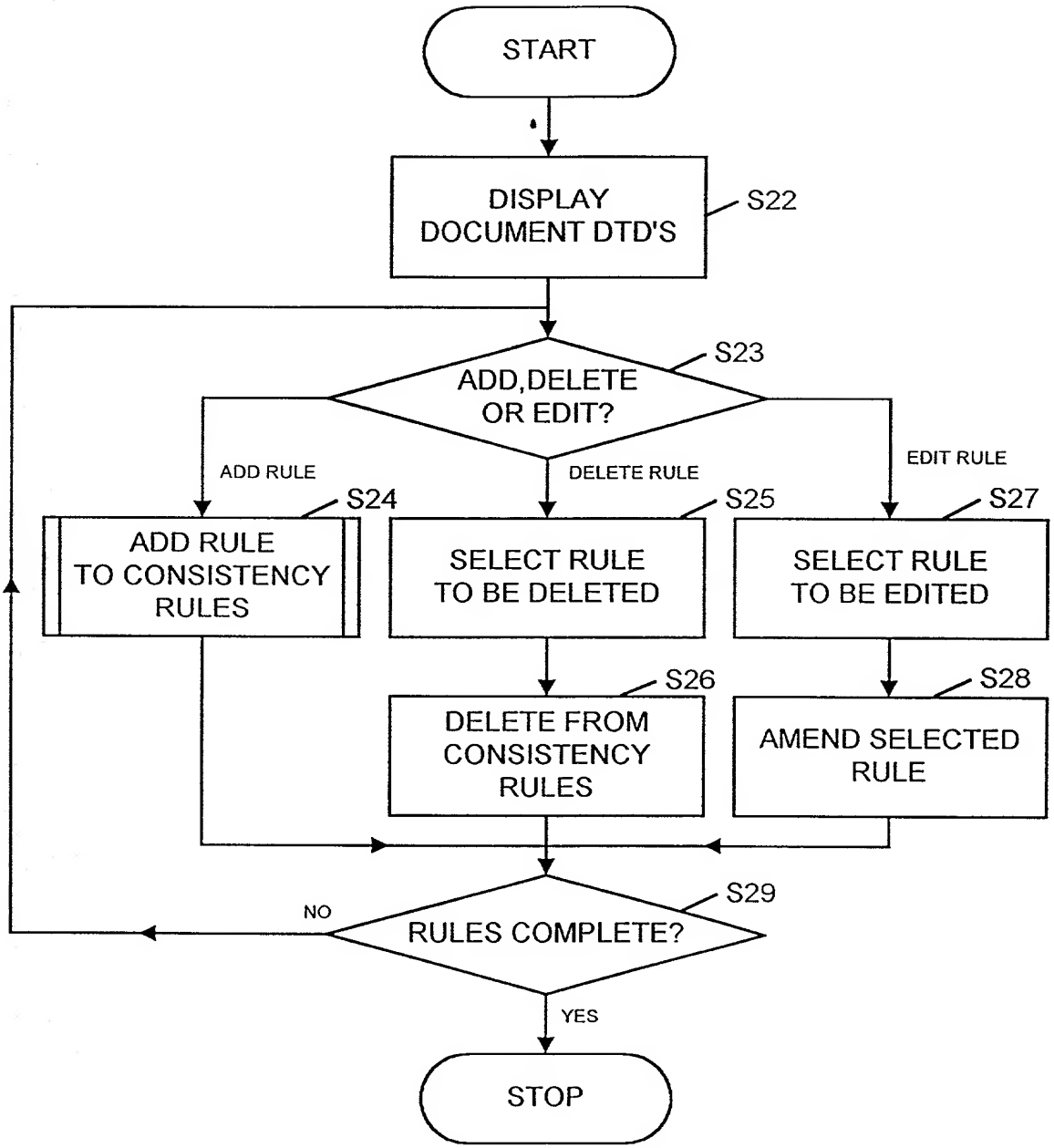


FIG. 9

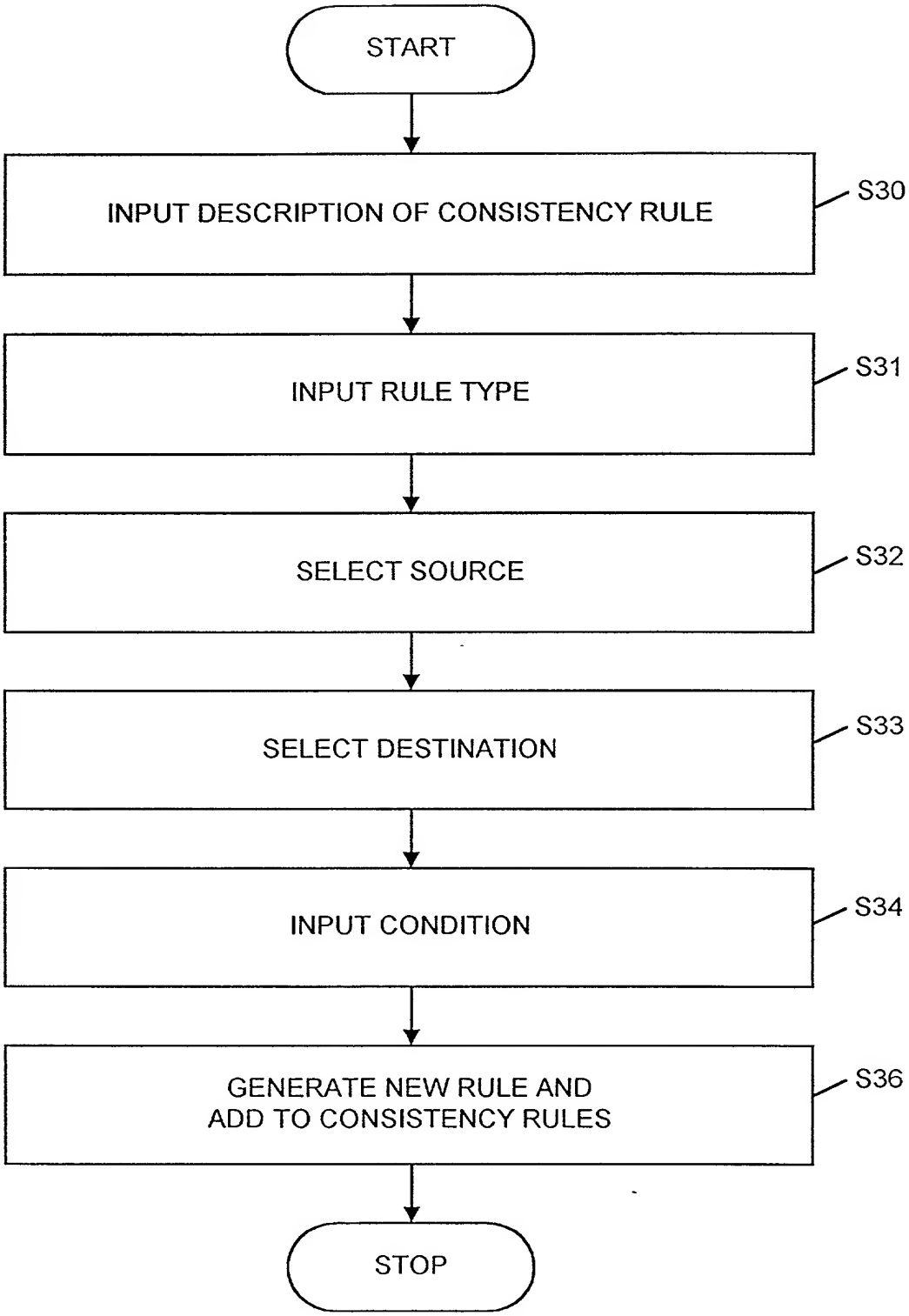


FIG. 10

10/009722 - 05/22/02

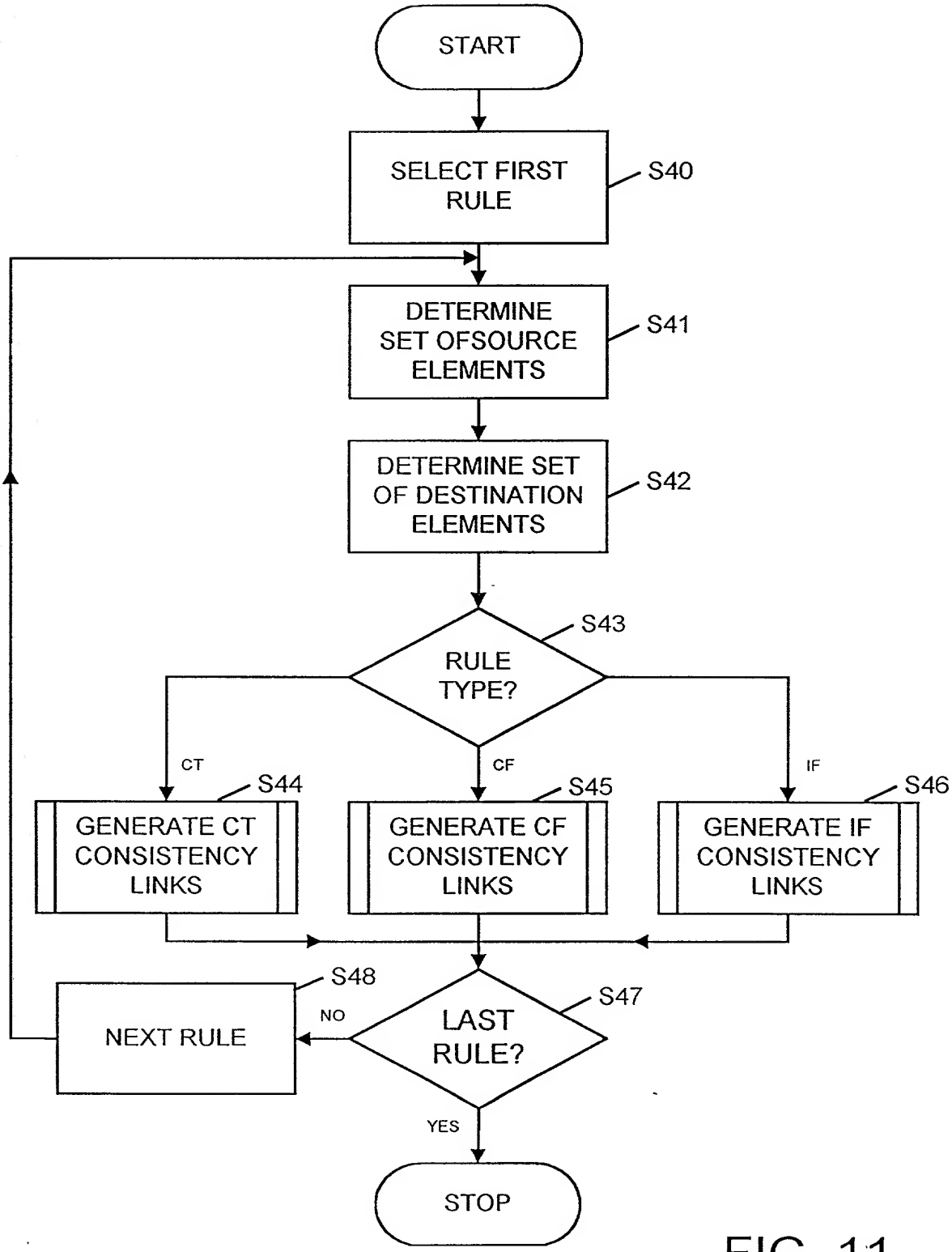

START
SELECT FIRST RULE
S40
DETERMINE SET OFSOURCE ELEMENTS
S41
DETERMINE SET OF DESTINATION ELEMENTS
S42
RULE TYPE?
S43
CT
CF
IF
GENERATE CT CONSISTENCY LINKS
S44
GENERATE CF CONSISTENCY LINKS
S45
GENERATE IF CONSISTENCY LINKS
S46
LAST RULE?
S47
NO
NEXT RULE
S48
YES
STOP

FIG. 11

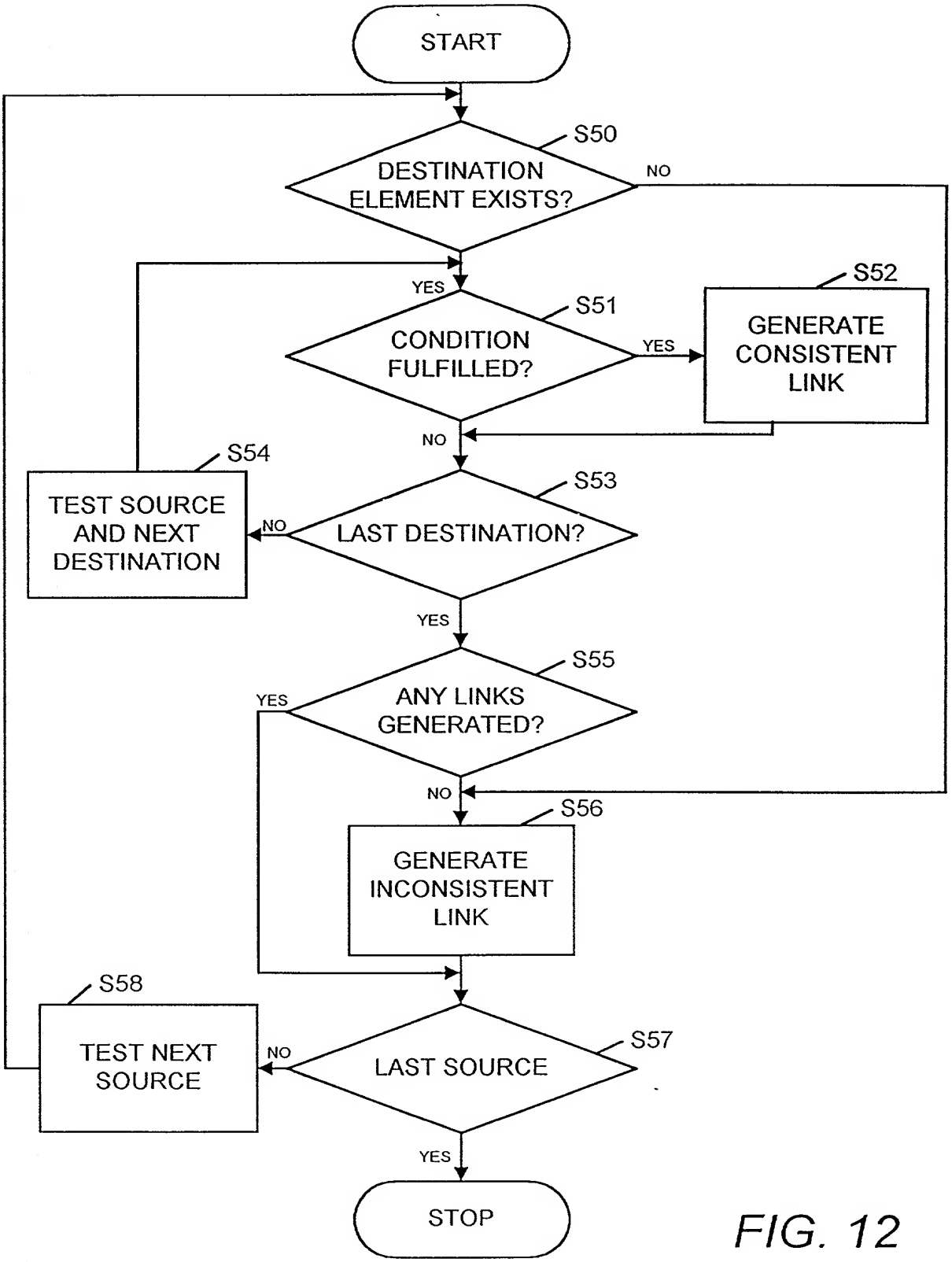


FIG. 12

10009722 052202

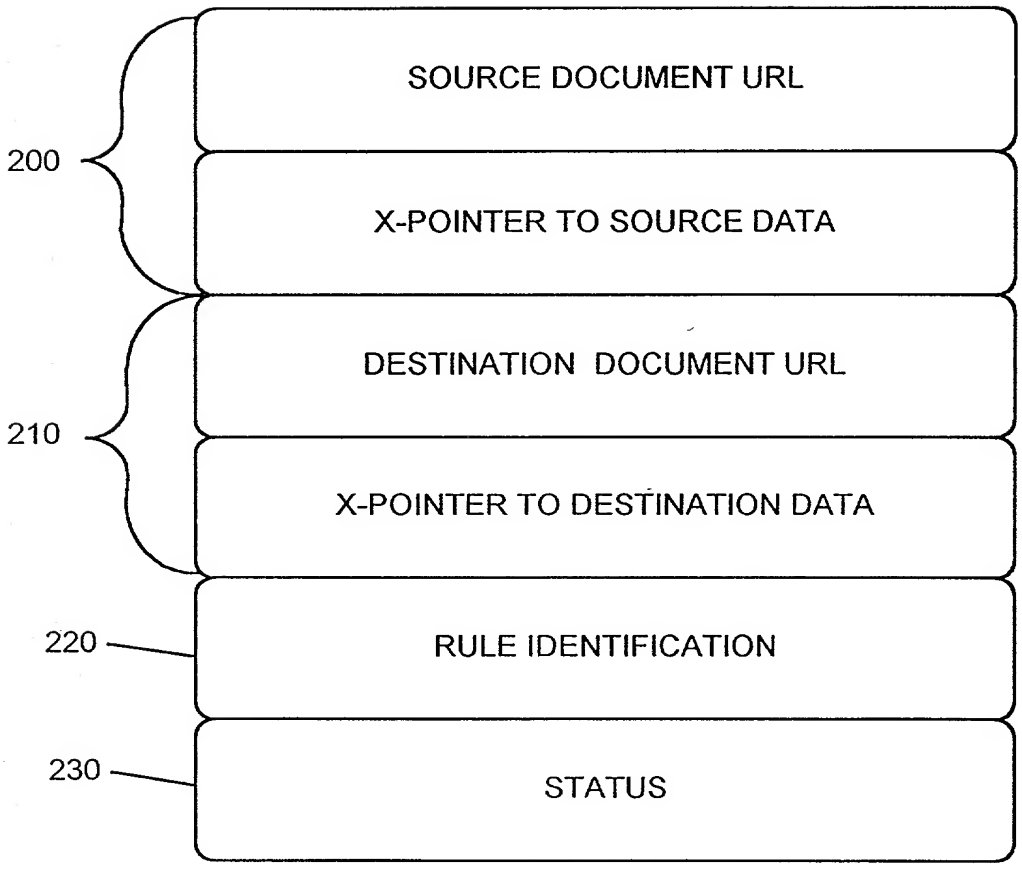


FIG. 13

10009722 052202

10/009722 JC05 Rec'd PCT/PTO 22 NOV 2001

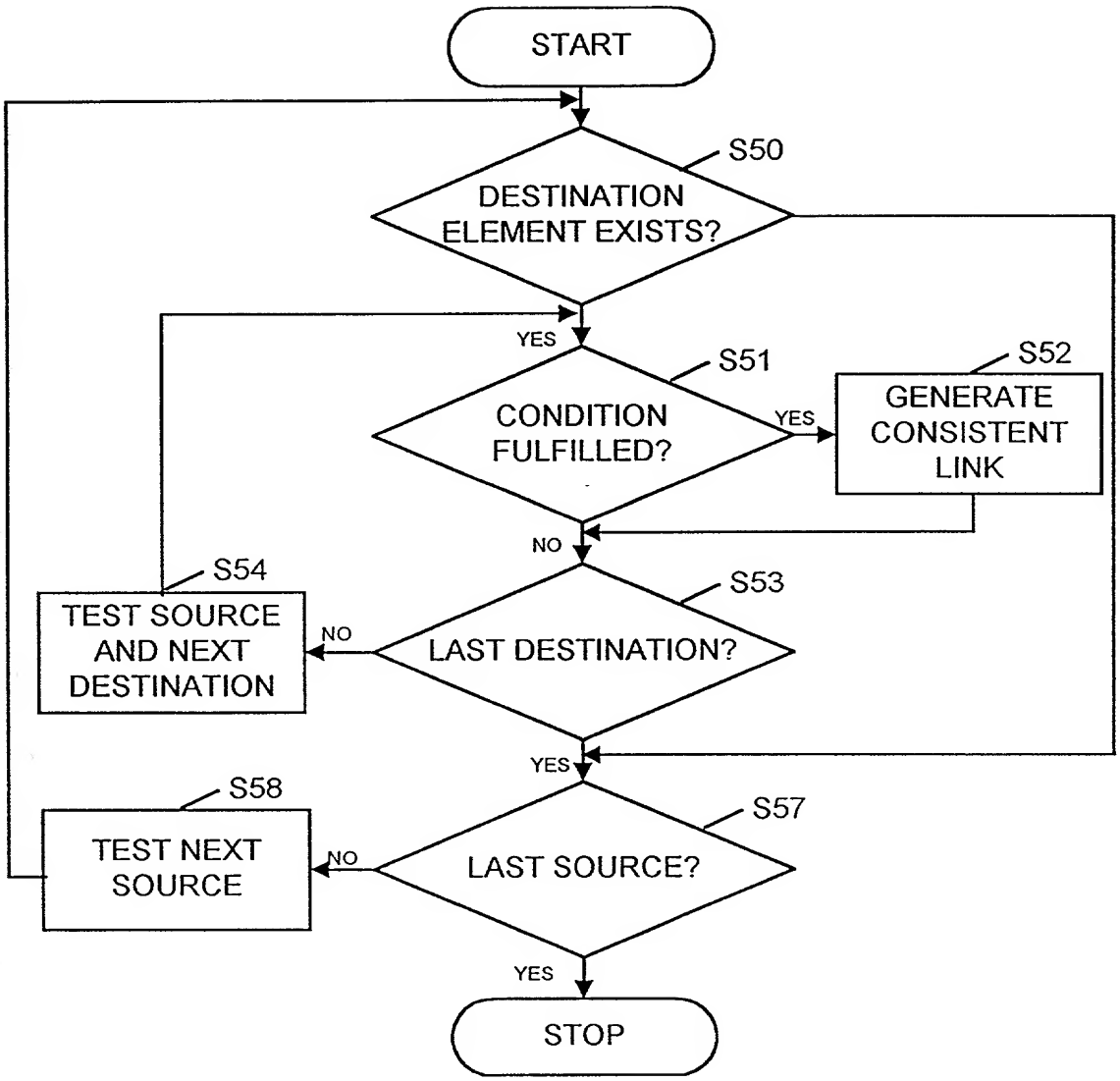


FIG. 14

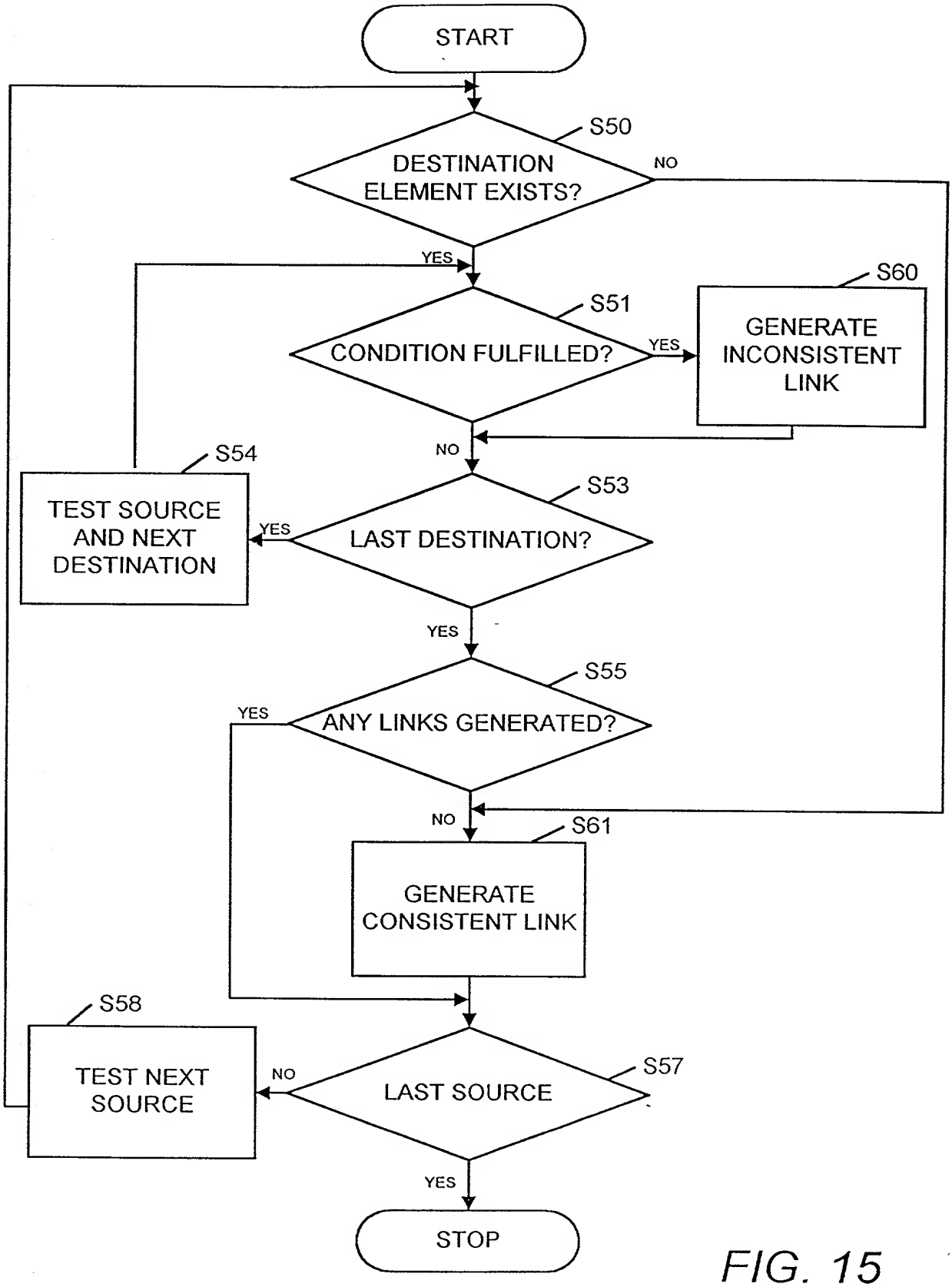


FIG. 15

10009722 052202

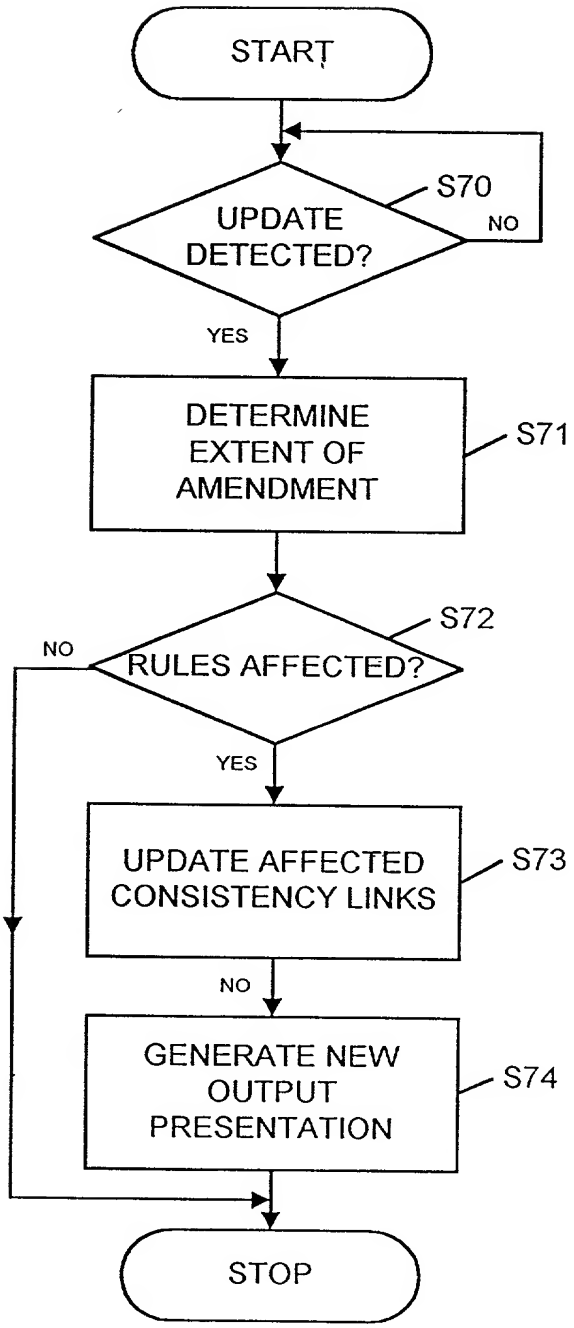


FIG. 16

10009722 052202

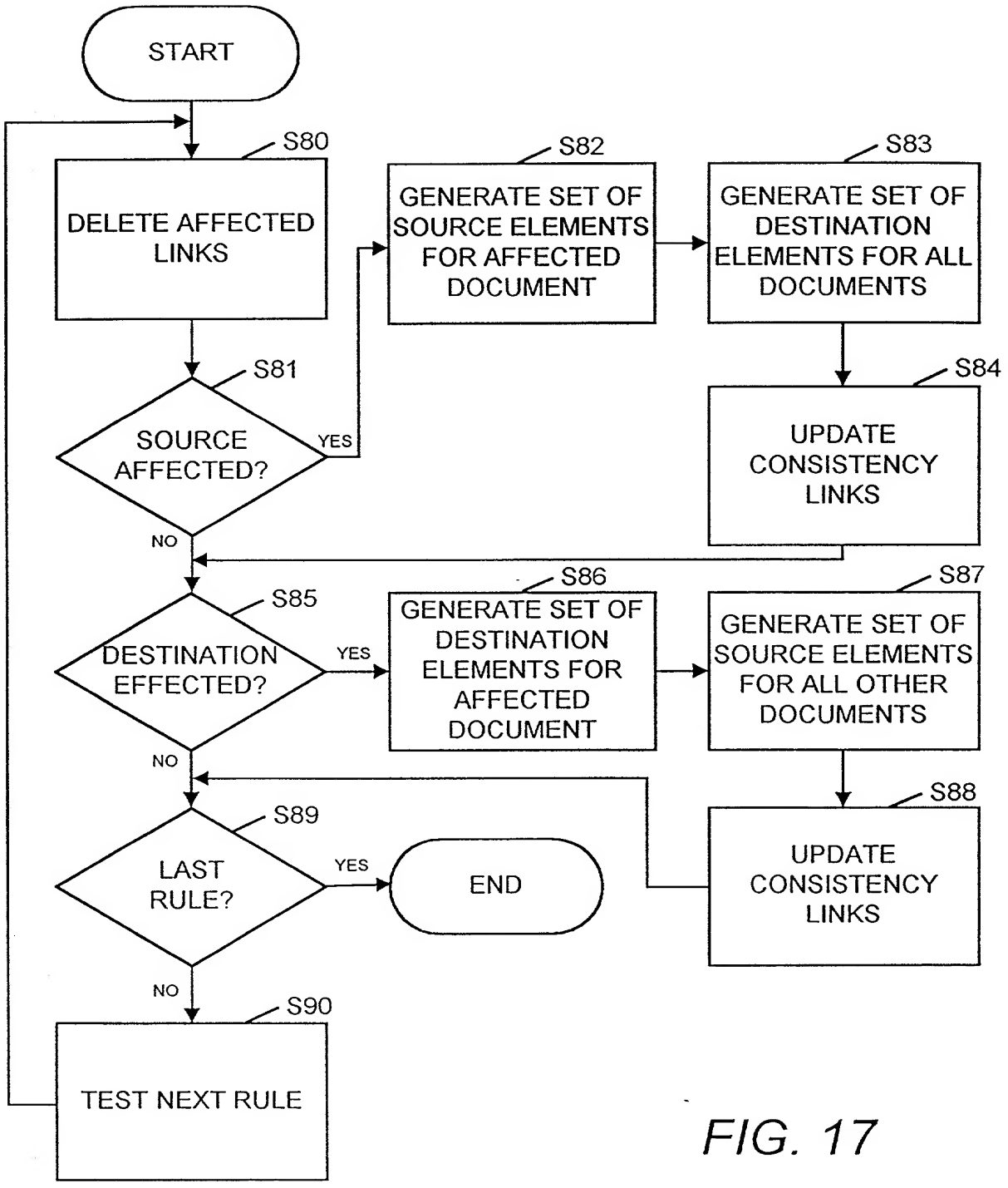

START
S80
DELETE AFFECTED LINKS
S81
SOURCE AFFECTED?
YES
NO
S82
GENERATE SET OF SOURCE ELEMENTS FOR AFFECTED DOCUMENT
S83
GENERATE SET OF DESTINATION ELEMENTS FOR ALL DOCUMENTS
S84
UPDATE CONSISTENCY LINKS
S85
DESTINATION EFFECTED?
YES
NO
S86
GENERATE SET OF DESTINATION ELEMENTS FOR AFFECTED DOCUMENT
S87
GENERATE SET OF SOURCE ELEMENTS FOR ALL OTHER DOCUMENTS
S88
UPDATE CONSISTENCY LINKS
S89
LAST RULE?
YES
NO
END
S90
TEST NEXT RULE

FIG. 17

Customer Number 22,852
Attorney Docket No. 08364.0031

DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that: my residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND APPARATUS FOR MONITORING AND MAINTAINING THE CONSISTENCY OF DISTRIBUTED DOCUMENTS

the specification of which
☐ is attached and/or
☒ was filed on December 17, 2001 as United States Application Serial No. 10/009722 or
☒ PCT International Application No. PCT/GB00/02336, and filed on June 15, 2000.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR § 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. § 119(a)-(d) or § 365(b) of any foreign application(s) for patent or inventor's certificate or § 365(a) of any PCT International application(s) designating at least one country other than the United States, listed below and have also identified below, any foreign application(s) for patent or inventor's certificate, or any PCT International application(s) having a filing date before that of the application(s) of which priority is claimed:

Country	Application Number	Date of Filing	Priority Claimed Under 35 U.S.C. 119
Great Britain	9914232.5	June 18, 1999	☒ YES ☐ NO
			☐ YES ☐ NO

I hereby claim the benefit under 35 U.S.C. § 119(e) of any United States provisional application(s) listed below:

Application Number	Date of Filing

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s) or § 365(c) of any PCT International application(s) designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application(s) in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR § 1.56 which became available between the filing date of the prior application(s) and the national or PCT International filing date of this application:

Application Number	Date of Filing	Status (Patented, Pending, Abandoned)

I hereby appoint the following attorney and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. **FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P., CUSTOMER NUMBER 22,852**, Douglas B. Henderson, Reg. No. 20,291; Ford F. Farabow, Jr., Reg. No. 20,630; Arthur S. Garrett, Reg. No. 20,338; Donald R. Dunner, Reg. No. 19,073; Brian G. Brunsvold, Reg. No. 22,593; Tipton D. Jennings, IV, Reg. No. 20,645; Jerry D. Voight, Reg. No. 23,020; Laurence R. Hefter, Reg. No. 20,827; Kenneth E. Payne, Reg. No. 23,098; Herbert H. Mintz, Reg. No. 26,691; C. Larry O'Rourke, Reg. No. 26,014; Albert J. Santorelli, Reg. No. 22,610; Michael C. Elmer, Reg. No. 25,857; Richard H. Smith, Reg. No. 20,609; Stephen L. Peterson, Reg. No. 26,325; John M. Romary, Reg. No. 26,331; Bruce C. Zotter, Reg. No. 27,680; Dennis P. O'Reilley, Reg. No. 27,932; Allen M. Sokal, Reg. No. 26,695; Robert D. Bajefsky, Reg. No. 25,387; Richard L. Stroup, Reg. No. 28,478; David W. Hill, Reg. No. 28,220; Thomas L. Irving, Reg. No. 28,619; Charles E. Lipsey, Reg. No. 28,165; Thomas W. Winland, Reg. No. 27,605; Basil J. Lewris, Reg. No. 28,818; Martin I. Fuchs, Reg. No. 28,508; E. Robert Yoches, Reg. No. 30,120; Barry W. Graham, Reg. No. 29,924; Susan Haberman Griffen, Reg. No. 30,907; Richard B. Racine, Reg. No. 30,415; Thomas H. Jenkins, Reg. No. 30,857; Robert E. Converse, Jr., Reg. No. 27,432; Clair X. Mullen, Jr., Reg. No. 20,348; Christopher P. Foley, Reg. No. 31,354; John C. Paul, Reg. No. 30,413; Roger D. Taylor,

59

Reg. No. 28,992; David M. Kelly, Reg. No. 30,953; Kenneth J. Meyers, Reg. No. 25,146; Carol P. Einaudi, Reg. No. 32,220; Walter Y. Boyd, Jr., Reg. No. 31,738; Steven M. Anzalone, Reg. No. 32,095; Jean B. Fordis, Reg. No. 32,984; Barbara C. McCurdy, Reg. No. 32,120; James K. Hammond, Reg. No. 31,964; Richard V. Burgujian, Reg. No. 31,744; J. Michael Jakes, Reg. No. 32,824; Thomas W. Banks, Reg. No. 32,719; Christopher P. Isaac, Reg. No. 32,616; Bryan C. Diner, Reg. No. 32,409; M. Paul Barker, Reg. No. 32,013; Andrew Chanho Sonu, Reg. No. 33,457; David S. Forman, Reg. No. 33,694; Vincent P. Kovalick, Reg. No. 32,867; James W. Edmondson, Reg. No. 33,871; Michael R. McGurk, Reg. No. 32,045; Joann M. Neth, Reg. No. 36,363; Gerson S. Panitch, Reg. No. 33,751; Cheri M. Taylor, Reg. No. 33,216; Charles E. Van Horn, Reg. No. 40,266; Linda A. Wadler, Reg. No. 33,218; Jeffrey A. Berkowitz, Reg. No. 36,743; Michael R. Kelly, Reg. No. 33,921; James B. Monroe, Reg. No. 33,971; Doris Johnson Hines, Reg. No. 34,629; Allen R. Jensen, Reg. No. 28,224; Lori Ann Johnson, Reg. No. 34,498; and David A. Manspeizer, Reg. No. 37,540 and . Please address all correspondence to **FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.**, 1300 I Street, N.W., Washington, D.C. 20005; Telephone No. (202) 408-4000.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Full Name of First Inventor	Inventor's Signature	Date
Andrea ZISMAN	Andrea Zisman	12-02-2002
Residence		Citizenship
London, United Kingdom GBX		Brazil
Post Office Address		
University College London, Department of Computer Science, Gower Street, London WC1E 6BT, United Kingdom		

Full Name of Second Inventor	Inventor's Signature	Date
Anthony FINKELSTEIN	A W Finkelstein	19-04-02
Residence		Citizenship
London, United Kingdom GBX		Great Britain
Post Office Address		
University College London, Department of Computer Science, Gower Street, London WC1E 6BT, United Kingdom		

Full Name of Third Inventor	Inventor's Signature	Date
Ernst ELLMER		4-2-02
Residence		Citizenship
London, United Kingdom GBX		Austria
Post Office Address		
University College London, Department of Computer Science, Gower Street, London WC1E 6BT, United Kingdom		

Full Name of Fourth Inventor	Inventor's Signature	Date
Danila SMOLKO	Danila Smolko	29-04-02
Residence		Citizenship
London, United Kingdom GBX		Russia
Post Office Address		
University College London, Department of Computer Science, Gower Street, London WC1E 6BT, United Kingdom		

Full Name of Fifth Inventor	Inventor's Signature	Date
Wolfgang EMMERICH		3-2-02
Residence		Citizenship
London, United Kingdom GBX		Germany
Post Office Address		
University College London, Department of Computer Science, Gower Street, London WC1E 6BT, United Kingdom		